

**RANCANG BANGUN *PROTOTYPE* SISTEM *REAL TIME*
PENDETEKSI PELANGGAR *ZEBRA CROSS* PADA *TRAFFIC*
LIGHT DENGAN MENGGUNAKAN ARDUINO**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Reza Ridlo Nugraha
NIM: 135150300111040



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

RANCANG BANGUN *PROTOTYPE* SISTEM *REAL TIME* PENDETEKSI PELANGGAR
ZEBRA CROSS PADA *TRAFFIC LIGHT* MENGGUNAKAN ARDUINO

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Reza Ridlo Nugraha

NIM: 135150300111040

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, S.T, M.Eng.

NIP: 19820809 201212 1 004

Gembong Edhi Setyawan, S.T., M.T.

NIK: 201208 761201 1 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Agustus 2018

Reza Ridlo Nugraha

NIM: 135150300111040



KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga peneliti dapat menyelesaikan penulisan skripsi dengan judul “Rancang Bangun *Prototype* Sistem *Real Time* Pendeteksi Pelanggar *Zebra Cross* Pada *Traffic Light* Dengan Menggunakan Arduino”.

Pada penulisan dan penelitian skripsi ini tidak lepas dari bantuan, bimbingan serta doa yang diberikan dari berbagai pihak. Dengan kerendahan hati peneliti mengucapkan terima kasih sebanyak-banyaknya kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Heru Nurwarsito, Ir., M.Kom. selaku Wakil Ketua I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. dosen pembimbing satu yang telah memberikan ilmu, saran, penjelasan dan motivasi serta membantu dalam penyusunan laporan penulis.
6. Bapak Gembong Edhi Setyawan, S.T., M.T. selaku dosen pembimbing dua yang telah memberikan ilmu, saran, penjelasan dan motivasi kepada penulis.
7. Seluruh civitas akademika Informatika Universitas Brawijaya dan terkhusus untuk teman-teman Teknik Komputer Angkatan 2013 yang telah banyak memberi bantuan dan dukungan selama peneliti menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian skripsi ini.
8. Bapak Eko, Bapak Asrofin, Ibu Wiwit dan Ibu Yulis selaku orang tua yang penulis cintai serta seluruh keluarga besar yang selalu memberi dukungan serta doa agar penulis dapat menyelesaikan skripsi ini dengan lancar.
9. Faviansyah, Tatit, Rizki, Elsadio, Ade, Noor, Adiet, Tegar, Alrynto, Arif, Iqbal, Guruh, Andre, Okky, Sonny, Nadea, Harfian, Kevin, Alfi dan teman-teman yang tidak dapat disebutkan satu persatu yang telah memberikan dukungan, kerjasama dan juga motivasi
10. Dan seluruh pihak yang tidak dapat disebutkan satu persatu, peneliti mengucapkan banyak terima kasih atas segala bentuk dukungan dan doa sehingga laporan skripsi ini dapat terselesaikan.

Peneliti menyadari bahwa tulisan ini masih jauh dari kata sempurna dan masih memiliki berbagai macam kekurangan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun, agar ke depannya penulis dapat menjadi lebih baik lagi. Akhir kata penulis berharap agar skripsi ini dapat bermanfaat bagi semua pihak dikemudian hari.

Malang, 1 Agustus 2018

Reza Ridlo Nugraha

rezaridlonugraha@gmail.com



ABSTRAK

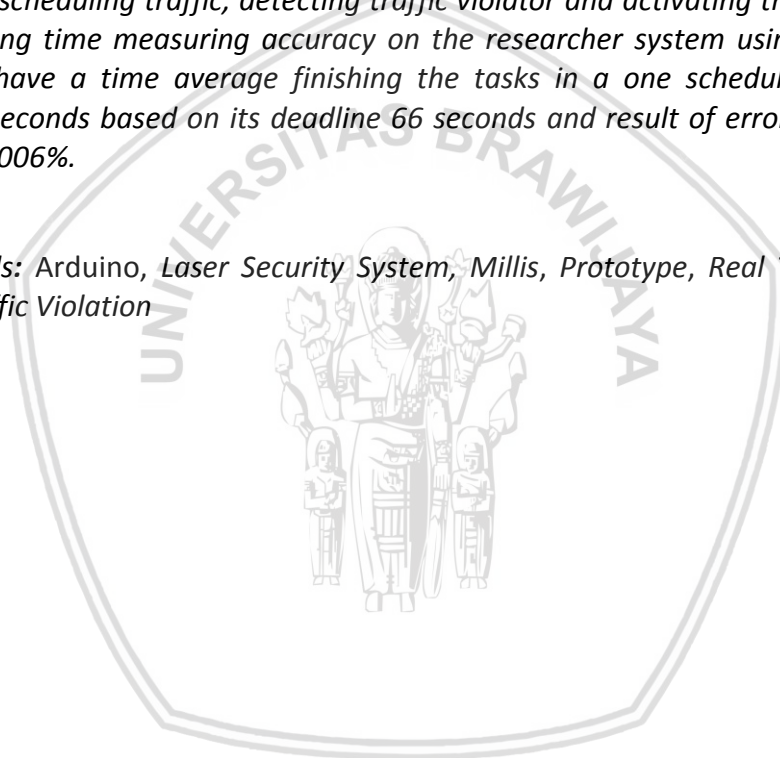
Banyaknya pengguna kendaraan pada zaman saat ini terutama di Indonesia tidak lepas dari banyaknya juga pelanggaran lalu lintas. Mikrokontroler Arduino merupakan komputer mini mikro yang dapat digunakan sebagai sistem kontrol pada perangkat yang saling terhubung dan berguna untuk membantu kebutuhan manusia diantaranya untuk membangun sistem otomatis maupun keamanan. Oleh karena ini peneliti akan merancang sebuah prototipe sistem pendeteksi pelanggaran pada lampu lalu lintas secara real time. sistem akan menggunakan konsep *laser security system* untuk mendeteksi pelanggaran, modul kamera VC0706 untuk mengambil data gambar dan menggunakan Millis untuk melakukan *multitasking* dan *scheduling* pada sistem agar berjalan secara real time. Pada Hasil pengujian fungsionalitasnya, sistem mampu menyelesaikan tugasnya yaitu melakukan penjadwalan lampu lalu lintas, mendeteksi pelanggaran dan mengaktifkan alarm. Pada pengujian ketepatan waktu pada sistem peneliti dengan menggunakan millis, sistem memiliki rata-rata waktu penyelesaian tugas dalam satu loop penjadwalan selama 66,004 detik berdasarkan *deadline* yaitu 66 detik dan memiliki persentase error ketepatan waktu sekitar 0,006%.

Kata kunci: Arduino, *Laser Security System*, Millis, Prototipe, *Real Time Sistem*, dan *Traffic Violation*

ABSTRACT

A lot of vehicles users nowadays especially in Indonesia can not be avoidable from a lot traffic violators on traffic light. Arduino microcontroller is a mini computer that can be used as a control system based on connected devices and Arduino used to build automatic system or security system. Therefore, the researcher will build a real time system prototype that can be used for detecting traffic violaton on traffic lights. System using the concept of laser security system to detect traffic violators, VC0706 camera modules to retrieve image data and using millis function to help multitask and scheduling on the systems to run in real time. In the results of its functionality testing, the system accomplishes its tasks of scheduling traffic, detecting traffic violator and activating the alarms. At the testing time measuring accuracy on the researcher system using millis, the system have a time average finishing the tasks in a one scheduling loop for 66.004 seconds based on its deadline 66 seconds and result of error percentage about 0.006%.

Keywords: *Arduino, Laser Security System, Millis, Prototype, Real Time Sistem, dan Traffic Violation*



DAFTAR ISI

PENGESAHAN	ii
PENYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah	3
1.5 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 <i>Real Time System</i>	6
2.2.1.1 Jenis <i>Real Time System</i> Berdasarkan <i>Deadline</i>	6
2.2.1.2 Karakteristik <i>Real Time System</i>	7
2.2.2 <i>Laser Security System</i>	7
2.2.3 Millis	8
2.2.4 Perhitungan Persentase Error	8
BAB 3 METODOLOGI	9
3.1 Diagram Alir	9
3.2 Studi Literatur	10
3.3 Analisis Kebutuhan.....	10
3.4 Perancangan Sistem dan Impelementasi.....	10
3.5 Pengujian dan Analisis Sistem.....	10

3.6 Penarikan Kesimpulan dan Saran.....	11
BAB 4 ANALISIS KEBUTUHAN	12
4.1 Kebutuhan Sistem	12
4.1.1 Kebutuhan Perangkat Keras.....	12
4.1.2 Kebutuhan Perangkat Lunak.....	19
4.2 Kebutuhan Fungsional	19
4.3 Kebutuhan Non Fungsional.....	20
4.3.1 Karakteristik Pengguna	20
4.3.2 Lingkungan Operasi.....	20
4.3.3 Asumsi dan Ketergantungan	20
4.3.4 Batasan Perancangan dan Implementasi.....	21
BAB 5 PERANCANGAN DAN IMPLEMENTASI	22
5.1 Perancangan Sistem.....	23
5.1.1 Perancangan Perangkat Keras	23
5.1.1.1 Perancangan Skematik Rangkaian LED	23
5.1.1.2 Perancangan Skematik Rangkaian Modul LDR dan Laser Diode	25
5.1.1.3 Perancangan Skematik Rangkaian Buzzer	26
5.1.1.4 Perancangan Skematik Rangkaian VC0706 dan Modul Micro SD	27
5.1.1.5 Perancangan Skematik Rangkaian Aktuator Servo SG90... ..	29
5.1.1.6 Perancangan Tata Letak dan Ukuran Prototipe.....	30
5.1.2 Perancangan Perangkat Lunak.....	31
5.2 Implementasi Sistem.....	33
5.2.1 Implementasi Perangkat Keras	33
5.2.2 Implementasi Perangkat Lunak.....	35
5.2.2.1 Implementasi Setup Sistem	35
5.2.2.2 Implementasi <i>Main Function</i>	37
5.2.2.3 Implementasi Fungsi Lampu	38
5.2.2.4 Implementasi Fungsi Servo	41
5.2.2.5 Implementasi Fungsi Deteksi	42
5.2.2.6 Implementasi Fungsi Kamera.....	44

BAB 6 PENGUJIAN DAN ANALISIS	46
6.1 Pengujian <i>Scheduling</i> Sistem	46
6.1.1 Tujuan Pengujian.....	46
6.1.2 Pelaksanaan Pengujian	46
6.1.3 Prosedur Pengujian.....	46
6.1.4 Hasil Pengujian.....	47
6.1.5 Analisis Pengujian	52
6.2 Pengujian Deteksi Pelanggar.....	52
6.2.1 Tujuan Pengujian.....	52
6.2.2 Pelaksanaan Pengujian	53
6.2.3 Prosedur Pengujian.....	53
6.2.4 Hasil Pengujian.....	53
6.2.5 Analisis Pengujian	64
6.3 Pengujian Waktu Eksekusi dalam satu siklus penjadwalan.....	65
6.3.1 Tujuan Pengujian.....	65
6.3.2 Pelaksanaan Pengujian	65
6.3.3 Prosedur Pengujian.....	65
6.3.4 Hasil Pengujian.....	65
6.3.5 Analisis Pengujian	67
BAB 7 PENUTUP	69
7.1 Kesimpulan.....	69
7.2 Saran	70
DAFTAR PUSTAKA	71

DAFTAR TABEL

Tabel 4.1 Spesifikasi Arduino MEGA 2560	13
Tabel 4.2 Spesifikasi Buzzer	14
Tabel 4.3 Spesifikasi Modul LDR	14
Tabel 4.4 Spesifikasi <i>Laser Diode</i>	15
Tabel 4.5 Spesifikasi VC0706 Camera	15
Tabel 4.6 Spesifikasi Modul Micro SD Card	16
Tabel 4.7 Spesifikasi Micro SD Card V-Gen	17
Tabel 4.8 Spesifikasi Servo SG90.....	17
Tabel 4.9 Spesifikasi Laptop yang digunakan.....	18
Tabel 5.1 Kabel Hubungan LED dengan Arduino	24
Tabel 5.2 Kabel Hubungan LDR dan <i>Laser Diode</i> dengan Arduino	25
Tabel 5.3 Kabel Hubungan Buzzer dengan Arduino	27
Tabel 5.4 Kabel Hubungan VC0706 dan Modul SD dengan Arduino	28
Tabel 5.5 Kabel Hubungan SG90 dengan Arduino	30
Tabel 6.1 Hasil Pengujian Penjadwalan pada Sistem.....	52
Tabel 6.2 Hasil Pengujian Fungsionalitas Deteksi Pelanggaran	64
Tabel 6.3 Data Rata-rata Waktu Eksekusi Alarm Pelanggar	64
Tabel 6.4 Data Waktu Eksekusi tiap loop sistem	68

DAFTAR GAMBAR

Gambar 2.1	Konsep Kerja <i>Laser Security System</i>	7
Gambar 3.1	Alur Metode Penelitian	9
Gambar 4.1	Diagram Analisis Kebutuhan.....	12
Gambar 5.1	Tahapan Perancangan dan Implementasi	22
Gambar 5.2	Skematik Rangkaian LED.....	23
Gambar 5.3	Skematik Rangkaian LDR dan <i>Laser Diode</i>	25
Gambar 5.4	Skematik Rangkaian Buzzer	26
Gambar 5.5	Skematik Rangkaian VC0706 dan Modul SD	27
Gambar 5.6	Skematik Rangkaian SG90	29
Gambar 5.7	Perancangan Tata Letak dan Ukuran Prototipe	31
Gambar 5.8	Diagram Alir Perancangan Perangkat Lunak	32
Gambar 5.9	Tampak Atas Perancangan Perangkat Keras	34
Gambar 6.1	Hasil pengujian pada 30 detik pertama.....	47
Gambar 6.2	Hasil pengujian pada detik ke 30 sampai detik ke 33	48
Gambar 6.3	Posisi Fokus Kamera pada detik ke 31.....	48
Gambar 6.4	Hasil pengujian pada detik ke 33 sampai detik ke 63	49
Gambar 6.5	Hasil pengujian pada detik ke 63 sampai detik ke 66	50
Gambar 6.6	Posisi Fokus Kamera pada detik ke 64.....	50
Gambar 6.7	Perbandingan waktu eksekusi sistem dengan waktu nyata	51
Gambar 6.8	Diagram Waktu Pengujian <i>Scheduling</i>	51
Gambar 6.9	Kondisi Sensor Deteksi tidak aktif	54
Gambar 6.10	Kondisi Sensor Deteksi aktif	54
Gambar 6.11	Hasil pengujian alarm deteksi	55
Gambar 6.12	Hasil pengujian pendeteksi pelanggaran dengan objek 1 menggunakan resolusi 640x480.....	56
Gambar 6.13	Hasil pengujian pendeteksi pelanggaran dengan objek 1 menggunakan resolusi 320x240.....	56
Gambar 6.14	Hasil pengujian pendeteksi pelanggaran dengan objek 1 menggunakan resolusi 160x120.....	57
Gambar 6.15	Hasil pengujian pendeteksi pelanggaran dengan objek 2 menggunakan resolusi 640x480.....	57

Gambar 6.16 Hasil pengujian pendeteksi pelanggaran dengan objek 2 menggunakan resolusi 320x240.....	58
Gambar 6.17 Hasil pengujian pendeteksi pelanggaran dengan objek 2 menggunakan resolusi 160x120.....	58
Gambar 6.18 Hasil pengujian pendeteksi pelanggaran dengan objek 3 menggunakan resolusi 640x480.....	59
Gambar 6.19 Hasil pengujian pendeteksi pelanggaran dengan objek 3 menggunakan resolusi 320x240.....	60
Gambar 6.20 Hasil pengujian pendeteksi pelanggaran dengan objek 3 menggunakan resolusi 160x120.....	60
Gambar 6.21 Hasil pengujian pendeteksi pelanggaran dengan objek 4 menggunakan resolusi 640x480.....	61
Gambar 6.22 Hasil pengujian pendeteksi pelanggaran dengan objek 4 menggunakan resolusi 320x240.....	61
Gambar 6.23 Hasil pengujian pendeteksi pelanggaran dengan objek 4 menggunakan resolusi 160x120.....	62
Gambar 6.24 Hasil pengujian pendeteksi pelanggaran dengan objek 5 menggunakan resolusi 640x480.....	62
Gambar 6.25 Hasil pengujian pendeteksi pelanggaran dengan objek 5 menggunakan resolusi 320x240.....	63
Gambar 6.26 Hasil pengujian pendeteksi pelanggaran dengan objek 5 menggunakan resolusi 160x120.....	63
Gambar 6.27 Hasil waktu eksekusi satu siklus program pada loop pertama	66
Gambar 6.28 Hasil waktu eksekusi satu siklus program pada loop kelima	66
Gambar 6.29 Hasil waktu eksekusi satu siklus program pada loop kesepuluh	67

BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada saat ini, teknologi selalu berkembang seiring berjalannya waktu. Hampir semua bidang mengalami kemajuan teknologi dalam bentuk apapun salah satunya ada otomatisasi dan juga keamanan. Mikrokontroler merupakan komputer mini mikro yang mempunyai masukan dan keluaran serta kendali pada alat yang terhubung dengannya dan program sesuai dengan kebutuhan. Mikrokontroler mengerjakan sesuatu perintah sesuai dengan instruksi-instruksi dari program yang tuliskan. Seiring perkembangannya mikrokontroler telah mengalami banyak perkembangan. Berbagai teknologi, fungsi yang diterapkan pada komponen ini menjadikan mikrokontroler yang saat ini memiliki banyak variasi. Dan sudah banyak pula aplikasi aplikasi yang dihasilkan oleh mikrokontroler, baik digunakan untuk membantu kebutuhan sehari-hari, dalam industri maupun inovasi teknologi. *Real Time System* merupakan sistem yang menghasilkan respon yang tepat dalam batas waktu yang ditentukan. Apabila respon dari sistem telah melewati batas waktu maka dapat menyebabkan kegagalan pada sistem atau penurunan perfromasi. Dengan kata lain *Real Time System* merupakan sistem yang tidak hanya mengutamakan eksekusi hasil kebenaran ekskusi namun juga hasil penyelesaian waktu eksekusi. Maka dari itu sistem harus memiliki fitur untuk memenuhi kebutuhan waktu dalam setiap kali menjalankan tugas. Sistem *real time* yang baik adalah sistem yang memiliki batas waktu eksekusi dimulainya atau berakhirnya sebuah tugas yang dapat di prediksi (Sagar, 2002).

Indonesia merupakan Negara yang memiliki banyak penduduk yang juga hampir setiap penduduknya merupakan pengguna kendaraan. Banyaknya pengguna jalan saat ini juga tidak lepas dari banyaknya pelanggar. Lampu Lalu Lintas merupakan sarana untuk mengatur lalu lintas pada jalan. Dimana pada lampu lalu lintas juga terdapat *zebra cross* sebagai sarana tempat untuk menyebrang jalan bagi pejalan kaki. *Zebra cross* yang memiliki fungsi yaitu memberikan prioritas bagi pejalan kaki untuk menyebrang jalan (Departemen Pekerjaan Umum, 1995).namun pada saat ini masih banyak pengemudi yang melanggar dengan melewati maupun berhenti tepat pada *zebra cross*. Sehingga dengan demikian, sarana untuk para pejalan kaki untuk menyebrang telah diambil oleh para pengguna kendaraan. Peraturan hukum mengenai *zebra cross* telah dijelaskan pada Undang-undang nomor 22 Tahun 2009 tentang lalu lintas dan angkutan jalan (LLAJ), pasal 131 Ayat (2), disebutkan bahwa "Pejalan kaki berhak mendapatkan prioritas pada saat menyebrang jalan di tempat penyebrangan" dan Pasal 106 Ayat (2), disebutkan bahwa "Setiap orang yang akan mengemudikan kendaraan bermotor di jalan wajib mengutamakan keselamatan pejalan kaki dan pesepeda". Kemudian dalam Pasal 284 disebutkan bahwa "setiap orang yang mengumudikan kendaraan bermotor dengan tidak mengutamakan keselamatan pejalan kaki atau pesepeda sebagaimana dimaksud

dalam Pasal 106 Ayat (2) dipidana dengan pidana kurungan paling lama 2 (dua) bulan atau denda paling banyak Rp. 500.000,00 (lima ratus ribu rupiah).

Pada penelitian ini penulis akan merancang sebuah sistem *prototype* yang menggunakan mikrokontroler yang berfungsi sebagai sarana pengatur ketertiban dan meminimalisir adanya pelanggaran dalam lingkup lampu lalu lintas terutama pada *zebra cross* secara *real time* dengan melakukan penjadwalan pada nyala lampu, servo serta memberikan *alarm* dan mengambil data atau gambar pada saat terjadi pelanggaran tepat sesuai masukan yang diberikan. Prototype ini terdiri dari menggunakan mikrokontroler Arduino MEGA sebagai pusat kendali kontrol aktuator dan sensor, LED sebagai indikasi lampu lalu lintas, LDR (*Light Dependent Resistor*) dan *Laser diode* sebagai pendeteksi pelanggar yang lewati *zebra cross* dengan menggunakan konsep *laser security system*, Buzzer sebagai *alarm* peringatan kepada pelanggar, serta Modul Kamera dan juga aktuator servo sebagai penggerak fokusnya. Bahasa pemrograman yang digunakan untuk merancang sistem ini adalah Arduino IDE. Sistem ini akan memanfaatkan fungsi Millis dalam melakukan *multitasking* dan *scheduling* agar sistem mampu bekerja secara *real time*.

1.2 Rumusan masalah

Berdasarkan latar belakang, terdapat beberapa rumusan permasalahan berdasarkan dari latar belakang diantaranya:

1. Bagaimana implementasi *real time system* pada sistem pendeteksi pelanggar pada *zebra cross* dengan menggunakan mikrokontroler arduino?
2. Bagaimana ketepatan hasil output sistem dalam melakukan pendeteksian pada pelanggar?
3. Bagaimana hasil ketepatan waktu eksekusi sistem untuk menyelesaikan *task* dengan menggunakan fungsi millis?

1.3 Tujuan

Adapun beberapa tujuan dari perancangan sistem, yaitu:

1. Untuk mengetahui sistem dapat bekerja sebagai penjadwal pada lampu lalu lintas serta pendeteksi pelanggaran pada *zebra cross* secara *real time*.
2. Untuk mengetahui ketepatan hasil output *alarm* yaitu buzzer dan kamera serta melakukan penyimpanan data pada saat terjadi sebuah pelanggaran secara tepat.
3. Untuk mengetahui ketepatan waktu eksekusi sistem untuk menyelesaikan *task* dengan menggunakan fungsi millis.

1.4 Manfaat

Adapun manfaat dari penelitian ini diantaranya:

1. Mengetahui seberapa rumit membangun suatu sistem pendeteksi pelanggaran *zebra cross* pada *traffic light* menggunakan mikrokontroler Arduino MEGA dengan menggunakan sensor dan aktuator.

2. Memberikan suatu hasil analisa tentang sistem mikrokontroler yang dapat melakukan *multitasking* maupun *scheduling*.
3. Agar Sistem dapat dikembangkan menjadi penelitian baru.

1.5 Batasan masalah

Adapun beberapa batasan-batasan masalah dari penelitian ini diantaranya:

1. Merancang *prototype* sistem kontrol menggunakan mikrokontroler Arduino MEGA.
2. Konsep yang digunakan untuk mendeteksi pelanggaran adalah *Laser Security System* yaitu dengan memanfaatkan sensor LDR dan *laser*.
3. Sistem melakukan *monitoring* dalam ruang lingkup dua persimpangan jalan.
4. *Multitasking* dan *scheduling* dilakukan menggunakan fungsi Millis.
5. Kamera yang digunakan untuk pengambilan data gambar adalah VC0706 *serial camera*

1.6 Sistematika pembahasan

Sistematika penulisan merupakan penjelasan umum bagian-bagian bab yang ada ada penelitian ini agar memudahkan pembaca untuk mengikuti alur pembahasan dalam penelitian. Sistematika penulisan akan dijelaskan sebagai berikut

BAB 1 PENDAHULUAN

Pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan, manfaat dan batasan masalah.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi kajian terhadap beberapa teori dan referensi yang menjadi landasan pendukung penelitian ini.

BAB 3 METODOLOGI

Pada bab ini berisi metode penelitian, metode pendekatan, metode pengumpulan data dan metode analisis data yang dilakukan pada penelitian ini.

BAB 4 ANALISIS KEBUTUHAN

Pada bab ini berisi tentang kebutuhan terkait penelitian yaitu kebutuhan perangkat keras, kebutuhan perangkat lunak serta kebutuhan fungsionalitas dan kebutuhan non fungsionalitas.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini berisi tentang bagaimana perancangan dan implementasi perangkat keras dan lunak yang digunakan.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini berisi pengujian dari implementasi yang dilakukan dan mengolah data yang di dapat untuk di analisa.

BAB 7 KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan dari analisa data yang didapatkan serta terdapat saran untuk penelitian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab ini membahas tentang pustaka yang digunakan sebagai acuan dalam penelitian. Beberapa teori dibahas menjadi acuan dalam perancangan penelitian ini. Terdapat beberapa sub bab yang akan dibahas diantaranya tinjauan pustaka yang merupakan kajian dari penelitian yang sudah dilakukan milik orang lain dan dasar teori yang menjelaskan tentang perangkat keras maupun lunak yang akan digunakan dalam penelitian.

2.1 Tinjauan Pustaka

Pada sub bab ini akan menjelaskan tentang perbandingan antara penelitian yang dilakukan dengan penelitian yang pernah dilakukan sebelumnya. Pada penelitian pertama telah melakukan penerapan pendeteksi pelanggaran pada lalu lintas dengan menggunakan sensor RFID reader serta RFID ID tag yang dipasangkan ke tiap kendaraan untuk pembacaan pelanggaran pada lalu lintas. serta dilengkapi dengan smart antenna yang tersambung dengan RFID reader untuk melakukan komunikasi jarak jauh. Pada saat kendaraan dengan tag melakukan sebuah pelanggaran maka reader akan membaca tag yang telah terpasang pada objek kendaraan lalu mengirimkan data violation packet sesuai dengan data pada tag kemudian menyimpannya dengan format *violation packet* ke database yang ada di dalam server (Mirzaei, 2013).

Pada rujukan yang kedua membahas tentang perancangan sebuah prototipe pendeteksi pelanggaran pada *traffic light* pada empat persimpangan jalan menggunakan dua mikrokontroler ATmega16 yang terhubung untuk menjalankan task nyala lampu serta pembacaan pelanggar dengan sensor photodiode. Sensor akan mengirimkan sinyal ke mikrokontroler lalu mengaktifkan webcam dengan aplikasi Delphi pada computer yang terhubung. Hasil Data gambar pelanggaran pada penelitian akan langsung ditampilkan melalui aplikasi Delphi yang terdapat pada PC (Wongsokuncoro, Winarno, & Akif, 2016).

Berdasarkan pada penelitian Mirzaei (2013) sistem dibuat lebih difokuskan pada pendataan pelanggar menuju ke server, penelitian yang dilakukan dengan memasang RFID tag untuk setiap kendaraan yang aktif sangat sulit untuk diimplementasikan secara langsung sehubungan dengan banyaknya pengguna kendaraan yang aktif. Sedangkan penelitian Wongsokuncoro, Winarno, & Akif (2016) merancang sebuah prototipe untuk mendeteksi pelanggaran pada *traffic light* dengan menggunakan mikrokontroler arduino lebih dari satu dalam melakukan *scheduling* dan *multitasking* pada sistem sehingga penelitian ini akan merancang sebuah prototipe pendeteksi pelanggaran tanpa menggunakan tag tetapi menggunakan sensor pendeteksi secara langsung dengan menggunakan konsep *laser security system* serta sistem dirancang menggunakan satu buah mikrokontroler saja. sistem dilengkapi dengan alarm pelanggaran yaitu buzzer dan kamera. Sistem diimplementasikan secara real time sehingga sistem dapat menjalankan beberapa *task* yang berjalan pada sistem bersamaan dan

melakukan penjadwalan task dengan *interval* waktu tertentu sesuai dengan *deadline* waktu yang diharapkan peneliti tanpa terhubung dengan perangkat lain yakni server maupun mikrokontroler lain. Dalam melakukan *Multitasking* dan *scheduling* pada sistem, peneliti akan memanfaatkan fungsi *Millis*.

2.2 Dasar Teori

Berikut merupakan beberapa dasar teori yang terkait pada penelitian ini.

2.2.1 Real Time System

Real Time System atau disebut sistem waktu nyata merupakan sistem yang menghasilkan respon yang tepat dalam batas waktu yang ditentukan. Apabila respon dari sistem telah melewati batas waktu maka dapat menyebabkan kegagalan pada sistem atau degradasi performansi. Sebuah *Real Time System* adalah sistem yang kebenarannya secara logis berdasarkan kebenaran hasil-hasil keluaran sistem dan ketepatan waktu tersebut dikeluarkan.

Real Time System diaplikasikan dalam konsep *multitasking* maupun *scheduling* pada sistem yang berjalan. Sistem real time tidak hanya mengutamakan ketepatan dalam menjalankan tugas-tugas bersamaan atau disebut juga *multitasking* namun juga memiliki *interval* waktu pada saat tugas tersebut akan dijalankan. Dengan kata lain sistem real time adalah sistem yang menggunakan *deadline* atau sistem yang dapat diprediksi batas waktu kapan tugas dikerjakan atau diselesaikan (Sagar, 2002). Berdasarkan batasan waktu yang dimiliki, *Real Time System* dibagi menjadi.

2.2.1.1 Jenis Real Time System Berdasarkan Deadline

Hard Real Time System

Hard Real Time System merupakan sistem yang harus menyelesaikan critical task pada waktu tertentu. Jika waktu kebutuhan tidak terpenuhi maka sistem akan gagal. Definisi lain disebutkan bahwa sistem dapat mentoleransi keterlambatan penyelesaian task tidak melebihi dari 100 *micro second*.

Soft Real Time System

Soft RealTime System merupakan sistem yang memiliki suatu batasan dalam menyelesaikan task. Jika waktu kebutuhan penyelesaian task tidak terpenuhi maka tidak akan berakibat fatal akan tetapi mengakibatkan penurunan performansi yang semakin menurun seiring waktu yang berjalan.

Firm Real Time System

Firm Real Time System merupakan sistem yang memiliki batasan sama seperti *Hard Real Time System*, namun sistem masih memberikan toleransi beberapa kali terhadap kegagalan dalam memenuhi batasan pada sistem, apabila jumlah kegagalan melewati batasan dari jumlah toleransi maka sistem akan gagal.

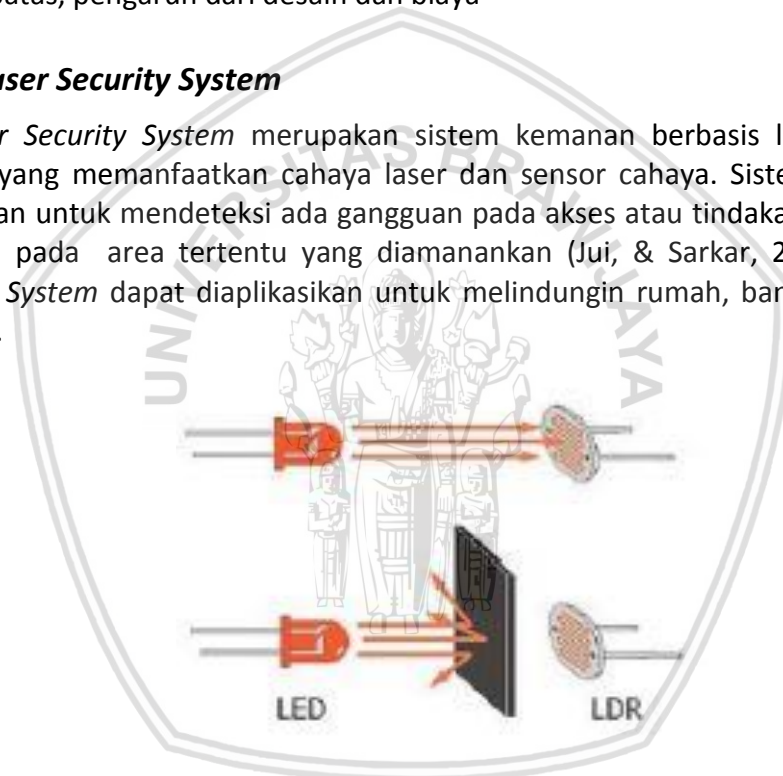
2.2.1.2 Karakteristik *Real Time System*

Terdapat pula beberapa karakteristik real time system diantaranya

1. *Realibility*, yaitu kemampuan sistem untuk bekerja tanpa bantuan manusia
2. *Predictability*, yaitu perilaku sistem yang dapat diprediksi dalam rentang waktu yang telah ditentukan
3. *Performance*, yaitu kemampuan sistem untuk menyelesaikan pekerjaan atau task secepat mungkin
4. *Compactness*, yaitu kemampuan sistem dalam penggunaan resource terbatas, pengaruh dari desain dan biaya

2.2.2 *Laser Security System*

Laser Security System merupakan sistem keamanan berbasis laser dengan konsep yang memanfaatkan cahaya laser dan sensor cahaya. Sistem keamanan digunakan untuk mendeteksi ada gangguan pada akses atau tindakan yang tidak diijinkan pada area tertentu yang diamankan (Jui, & Sarkar, 2009). *Laser Security System* dapat diaplikasikan untuk melindungi rumah, bank, loker dan lain-lain.



Gambar 2.1 Konsep Kerja *Laser Security System*

Sumber: (Jui, & Sarkar, 2009)

Berdasarkan Gambar 2.1 Konsep kerja dari *Laser Security System* adalah pembacaan nilai cahaya oleh sensor cahaya LDR, apabila sensor LDR membaca pada kondisi yang gelap, terdapat objek yang mengganggu atau tidak ada cahaya yang masuk maka nilai resistansinya akan semakin tinggi. Dengan kata lain dengan pemberian nilai *default* cahaya oleh laser ke sensor LDR apabila terdapat objek yang melewati atau mengganggu pembacaan nilai sensor LDR maka simpulkan terjadinya sebuah pelanggaran.

Pada kasus penelitian ini konsep *Laser Security System* akan digunakan untuk mendeteksi gangguan pelanggaran yang terjadi pada lampu lalu lintas terutama pada *zebra cross*.

2.2.3 Millis

Millis merupakan fungsi pada arduino IDE untuk perhitungan milli detik pada saat sistem mulai dijalankan. Millis memiliki tipe data *unsigned long*. Millis biasanya digunakan untuk melakukan perhitungan waktu yang berjalan pada sistem. Millis juga dapat dimanfaatkan sebagai *multitasking* dan *scheduling* pada mikrokontroler arduino untuk menjalankan beberapa task secara bersamaan maupun dengan waktu yang berbeda. Konsep kerja *multitasking* dan *scheduling* Millis adalah dengan mencatat waktu yang sedang berjalan dikurangi dengan waktu task yang dikerjakan sebelumnya kemudian dibandingkan dengan *interval* waktu pengerjaan tiap *task* yang telah ditentukan (Earl, 2014).

2.2.4 Penghitungan Persentase Error

Nilai Persentase Error suatu pengerjaan *deadline* pada sistem dapat dihitung menggunakan persamaan 2.1

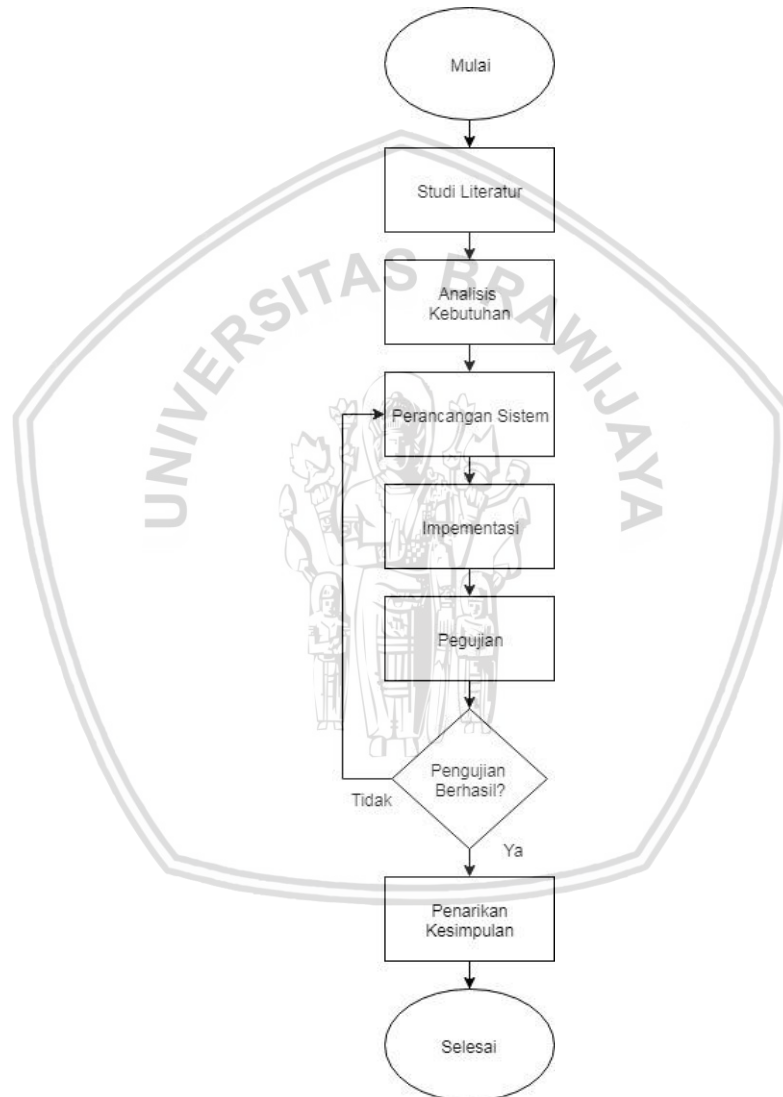
$$\text{Nilai Persentase Error} = \frac{\text{Rata-rata Error Waktu Penyelesaian Sistem}}{\text{Deadline}} \times 100\% \quad (2.1)$$

Nilai Persentase Error diperoleh dari nilai rata-rata waktu error yang didapat dari rata-rata waktu pengerjaan *task* satu dalam satu siklus dikurangi dengan nilai *deadline* kemudian dibagi dengan nilai *deadline* waktu nyata pengerjaan *task* dalam satu siklus yang telah ditentukan pada sistem dalam hitungan detik atau *second*.

BAB 3 METODOLOGI

3.1 Diagram Alir

Dalam mendesain sebuah sistem terdapat proses-proses yang saling terhubung agar suatu sistem menjadi terstruktur. Pada bab ini akan menjelaskan metode yang akan digunakan serta langkah yang dilakukan dalam penelitian ini. Tahapan pada penelitian ini digambarkan dalam bentuk diagram alir pada Gambar 3.1.



Gambar 3.1 Alur Metode Penelitian

Diagram alir Gambar 3.1 memiliki beberapa tahapan proses yaitu dimulai dari Studi literatur, analisis kebutuhan, Perancangan sistem, implementasi alat, lalu pengujian dengan dua hasil, jika pengujian sistem berhasil maka akan dilakukan penarikan kesimpulan. Jika pengujian tidak berhasil maka akan kembali pada tahapan perancangan. Penjelasan alur metode penelitian akan dijelaskan pada sub bab dibawah.

3.2 Studi Literatur

Pada bagian studi literature akan mempelajari mengenai pengertian dan penjelasan dari dasar teori yang digunakan dalam penulisan. Dasar teori tersebut diperoleh dari jurnal, forum, artikel, e-book, dokumentasi, ataupun buku. Teori-teori pada penelitian ini diantaranya meliputi : *Real Time System*, *Laser Security System*, Millis sebagai *multitasking* dan *scheduling* pada Sistem Mikrokontroler.

3.3 Analisis Kebutuhan

Analisis kebutuhan memiliki tujuan agar mendapatkan semua kebutuhan yang diperlukan dalam membangun sistem. Analisis kebutuhan dilakukan dengan mengidentifikasi apa saja yang dibutuhkan dan terlibat dalam suatu sistem yang akan dibuat. Pada kebutuhan sistem akan dianalisis apa saja yang diperlukan dalam pelaksanaannya seperti perangkat keras dan perangkat lunak. Kebutuhan perangkat keras akan membahas perangkat keras apa saja yang dibutuhkan secara spesifik, sedangkan kebutuhan perangkat lunak akan menjelaskan tentang perangkat lunak apa saja yang diperlukan dalam membantu perancangan sistem.

Pada kebutuhan *funksional* akan dijelaskan kesesuaian input yang dibutuhkan serta output yang akan dihasilkan pada sistem. Sedangkan pada kebutuhan *non-fungsional* akan membahas tentang lingkungan operasi asumsi dan ketergantungan, serta batasan perancangan dan implelementasi. Akan dibahas lebih rinci pada bab 4 analisis kebutuhan.

3.4 Perancangan Sistem Dan Implementasi

Pada tahapan ini bertujuan untuk merancang sistem agar kebutuhan *funksional* terpenuhi. Perancangan pada sistem dibagi menjadi perancangan perangkat keras yang saling terhubung untuk membangun sistem serta tata letak perangkat pada sistem. Dan juga perancangan perangkat lunak yang menjelaskan diagram alir logika pemograman yang akan dibangun.

Setelah tahapan perancangan telah selesai, akan dilanjutkan pada proses implementasi sistem, pada tahapan awal akan menjelaskan tentang implementasi perangkat keras yang merupakan gambaran dari sistem hasil rancangan perangkat keras sudah selesai dibangun. Lalu melakukan implementasi perangkat lunak yaitu mengimplementasikan source code yang telah dibuat agar sistem dapat berfungsi sesuai dengan kebutuhan *funksional* dari sistem. Arduino IDE digunakan untuk melakukan pemograman pada sistem.

3.5 Pengujian dan Analisis Sistem

Dalam tahap pengujian dan analisis sistem untuk menguji apakah sistem dibuat sesuai dengan yang diharapkan oleh peneliti. Terdapat beberapa parameter yang digunakan dalam pengujian sistem diantaranya pengujian *funksional* dari sistem. Untuk mengetahui sistem sudah berjalan sesuai dengan keinginan peneliti baik dalam melakukan penjadwalan *task*, kemampuan sistem untuk mendeteksi pelanggaran dan melakukan perbandingan rata-rata lama

waktu eksekusi satu siklus program pada sistem peneliti dengan menggunakan Millis dibandingkan dengan metode pengerjaan yang lain.

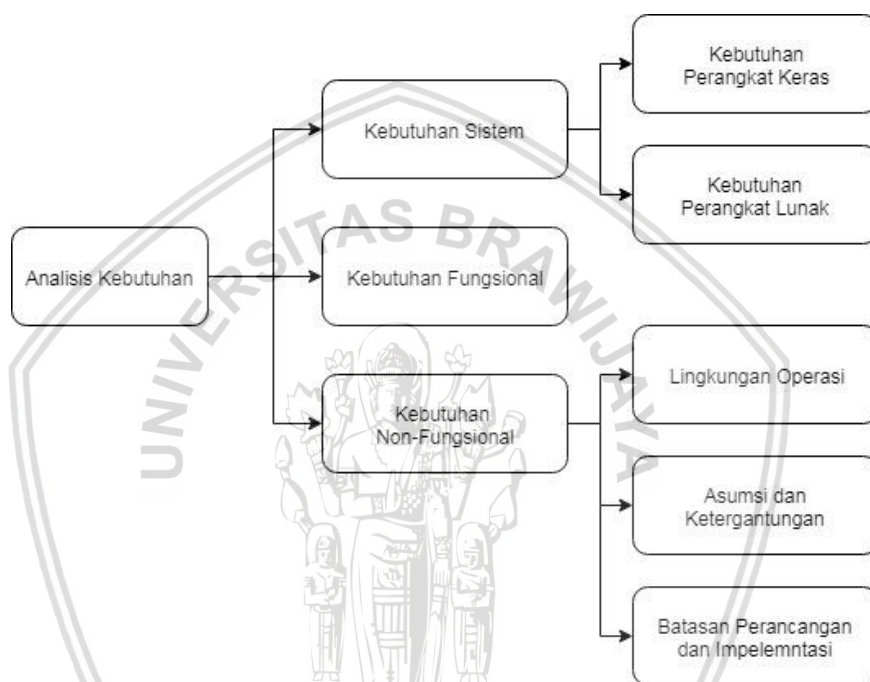
3.6 Penarikan Kesimpulan dan Saran

Kesimpulan akan didapatkan setelah melakukan seluruh tahap dari perancangan sistem, implementasi alat, pengumpulan hasil data, dan analisis hasil data. Kesimpulan diambil berdasarkan dari hasil pengujian dan analisis yang telah dilakukan. Isi dari kesimpulan dituliskan secara jelas, singkat dan memiliki tujuan untuk acuan pada penelitian yang mengembangkan penelitian yang berhubungan dengan penelitian ini. Selain itu, pada akhir penulisan juga terdapat saran yang bertujuan memperbaiki kesalahan yang terjadi sehingga dapat menyempurnakan penulisan serta untuk memberikan suatu pertimbangan dalam pengembangan pada sistem berikutnya.



BAB 4 ANALISIS KEBUTUHAN

Pada bab rekayasa kebutuhan akan menjelaskan tentang kebutuhan-kebutuhan yang diperlukan untuk merancang Sistem *Prototype* Pendeteksi Pelanggar *Zebra Cross*. Terdapat beberapa sub bab diantaranya kebutuhan sistem diantaranya kebutuhan perangkat keras, kebutuhan perangkat lunak, kebutuhan *fungsi*ional dan kebutuhan *non fungsi*ional. Pada kebutuhan *non fungsi*ional yang akan dibagi lagi menjadi lingkungan operasi, asumsi dan ketergantungan, serta batasan perancangan dan implementasi. Pada Gambar 4.1 akan menjelaskan tentang diagram analisis kebutuhan.



Gambar 4.1 Diagram Analisis Kebutuhan

4.1 Kebutuhan Sistem

Pada sub bab kebutuhan sistem berisi penjelasan tentang kebutuhan perangkat keras dan lunak sesuai dengan kebutuhan dalam merancang *Prototype* Pendeteksi Pelanggar *Zebra Cross*.

4.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras merupakan penjelasan tentang perangkat keras yang akan digunakan untuk membangun sistem. Kebutuhan perangkat keras akan dijelaskan diantaranya.

1. Arduino MEGA 2560

Merupakan mikrokontroler *single board* bersifat *open source* dan *softwrenya* mudah dikembangkan. Arduino Mega digunakan sebagai komponen pusat kontrol utama dari perangkat keras lain yang terhubung

yang juga berfungsi sebagai pemrosesan data input maupun output, menggerakkan aktuator dan juga pembacaan sensor. Arduino MEGA memiliki jumlah 54 pin digital input/output dan 16 pin analog output. Serta memiliki flash memory sebesar 256KB dan juga RAM sebesar 8KB. Berikut tabel 4.1 merupakan spesifikasi dari mikrokontroler Arduino MEGA yang digunakan.

Tabel 4.1 Spesifikasi Arduino MEGA 2560

Kategori	Spesifikasi
Mikrokontroler	ATMEGA 2560
Tegangan Pengoprasian	5V
Tegangan Input yang disarankan	7 – 12V
Batas Tegangan Input	6 – 20V
Jumlah pin I/O digital	54pin
Jumlah pin input Analog	16pin
Memori Flash	256KB
SRAM	8KB
EPROM	4KB
Clock Speed	16MHz

Berdasarkan spesifikasi pada tabel 4.1 pada penelitian ini menggunakan Arduino Mega karena memiliki spesifikasi yang lebih besar dibandingkan dengan jenis mikrokontroler arduino jenis lainnya. Salah satunya yaitu memiliki pin-pin yang lebih banyak yang akan digunakan untuk melakukan komunikasi antar perangkat keras yang terhubung. Pada sistem peneliti membutuhkan jumlah pin yang digunakan sebanyak 22 yang tidak bisa dipenuhi kebutuhannya apabila menggunakan mikrokontroler yang memiliki spesifikasi jumlah pin yang lebih rendah dibandingkan dengan Arduino Mega salah satunya adalah menggunakan Arduino Uno.

Arduino Mega memiliki *memory flash* yang digunakan untuk menyimpan program dan SRAM yang digunakan untuk menyimpan data baik variable maupun array yang cukup besar di dibandingkan jenis mikrokontroler arduino lainnya. Sehingga memungkinkan untuk membuat sebuah sistem yang menggunakan mikrokontroler yang lebih kompleks.

2. Buzzer

Merupakan komponen elektronika yang merubah sinyal listrik menjadi suara yang berfungsi sebagai *alarm* suara pada saat terjadi suatu kesalahan. Buzzer bekerja pada tegangan 5V. Pada penelitian buzzer akan digunakan sebagai *alarm* pada saat terjadi *trigger* pelanggaran pada suatu blok lampu lalu lintas. Frekuensi suara dan juga durasi alarm dapat disesuaikan dengan

kebutuhan pada sistem yang akan dibuat menggunakan fungsi *Tone*. Tabel 4.2 akan menjelaskan tentang spesifikasi buzzer yang digunakan.

Tabel 4.2 Spesifikasi Buzzer

Kategori	Spesifikasi
Tegangan Pengoperasian	5V
Arus Maksimal	30mA / 5vDC
Kekuatan Suara Maksimal	85dB
Frekuensi Resonansi	2500 +/- 300hz
Warna	Hitam

3. Modul LDR

Komponen yang berfungsi sebagai alat untuk mengukur intensitas cahaya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Modul LDR memiliki *digital output* dan juga dapat mengatur sensitifitas cahaya. Pada penelitian ini akan menggunakan nilai *digital output* dari modul LDR untuk mendeteksi pelanggaran dengan menggunakan konsep *laser security system*. Sistem menggunakan dua buah modul LDR untuk mendeteksi pelanggaran pada dua blok lampu lalu lintas. Tabel 4.3 akan menjelaskan tentang spesifikasi buzzer yang digunakan.

Tabel 4.3 Spesifikasi Modul LDR

Kategori	Spesifikasi
Tegangan Operasi	3.3V-5V
Output	Digital atau Analog
Sensitifitas	Adjustable

Berdasarkan spesifikasi pada tabel 4.3 penelitian ini menggunakan Modul LDR untuk memanfaatkan nilai *digital output* yaitu 0 dan 1 dari modul LDR untuk mendeteksi pelanggaran dengan membaca nilai pada LDR menjadi '1' atau pembacaan cahaya menurun. Sensifitas pembacaan nilai pada modul dapat diatur sesuai kebutuhan sistem berdasarkan parameter nilai cahaya yang didapatkan dengan *laser diode*.

4. Laser Diode

Merupakan komponen semikonduktor yang dapat menghasilkan radiasi yang dapat dilihat ataupun dalam bentuk spektrum infra merah (*Infrared/IR*) ketika dialiri arus listrik. Memiliki ukuran diameter sebesar 6mm. berfungsi sebagai pemberi nilai cahaya tetap atau *default* untuk modul LDR. Penelitian

menggunakan dua buah laser diode yang akan dipasangkan dengan dua buah modul LDR pada sistem.

Tabel 4.4 Spesifikasi Laser Diode

Kategori	Spesifikasi
Tegangan Operasi	5V
Ukuran	Diameter 6mm
Bentuk Spot	<i>DOT</i> atau titik
Suhu Kerja	-36 ~ +65 C

5. *Tradional Inorganic LED*

Merupakan komponen lampu cahaya yang sederhana pada penelitian ini digunakan sebagai suatu indikasi. Pada penelitian penggunaan LED digunakan sebagai indikasi lampu lalu lintas yang menyala bergantian sesuai dengan penjadwalan. Dan juga sebagai indikasi proses kamera pada saat sedang aktif. Jumlah LED yang digunakan pada penelitian sebanyak delapan buah LED. Enam buah LED dengan warna sepasang merah, kuning dan hijau digunakan sebagai indikasi pada dua blok lampu lalu lintas. Dan dua buah LED warna putih digunakan untuk indikasi pembacaan pada modul kamera pada saat aktif dan modul SD pada saat tidak berfungsi.

6. Modul Kamera VC0706

Merupakan modul kamera yang bekerja pada tegangan 5V berfungsi untuk pengambilan data gambar. VC0706 mempunyai kualitas standar VGA dan 2 output yaitu NTSC video dan *snapshot* yang datanya bisa dikirim menggunakan komunikasi serial TTL 3.3V. selain itu VC0706 mempunyai 3 ukuran resolusi gambar yang bisa diatur. Untuk spesifikasi selengkapnya akan dijelaskan pada tabel 4.5.

Tabel 4.5 Spesifikasi VC0706 Camera

Kategori	Spesifikasi
Tegangan Operasi	5V
Ukuran Modul	32mm x 32mm
<i>Image sensor</i>	CMOS ¼ inch
<i>Output Format</i>	<i>Standard</i> JPEG
<i>Frame speed</i>	640 x 480 30fps
<i>Image size</i>	VGA (640 x 480), QVGA (320 x 240), QQVGA (160 x 120)

<i>Baudrate</i>	9600 to 115200, default as 38400
Komunikasi	3.3V TTL

Berdasarkan spesifikasi pada tabel 4.5 modul kamera VC0706 mampu menyimpan data gambar dalam format .jpeg dan memiliki format ukuran gambar diantaranya 160x120, 320x240 dan paling besar sebesar 640x480 pixel. Memiliki setting *baudrate* dengan *default* 38400 yang mampu ditingkatkan paling tinggi menjadi 115200 untuk melakukan kecepatan data yang dikirim. Sistem akan memanfaatkan fungsi *snapshot* pada modul kamera VC0706 untuk digunakan sebagai hasil data pelanggaran yang terdeteksi oleh sensor pelanggar. Setting konfigurasi yang diperlukan oleh modul kamera VC0706 pada Arduino untuk mengaktifkan fungsi kamera jauh lebih sederhana dibandingkan dengan menggunakan modul kamera arduino lainnya dan penggunaannya lebih cocok digunakan untuk merancang sistem keamanan dibandingkan fotografi.

7. Modul Micro SD Card

Modul Micro SD Card merupakan komponen yang digunakan untuk menulis maupun membaca pada micro sd card. Pada penelitian modul Micro SD Card akan menuliskan data gambar yang didapat oleh modul kamera VC0706 menuju micro sd card mini yang terpasang pada modul tersebut. Modul menggunakan Komunikasi SPI untuk melakukan pembacaan atau penulisan data. Modul dapat melakukan penulisan data pada penyimpanan hanya pada micro SD Card dengan format PC FAT16/32. Tabel 4.6 akan menjelaskan tentang spesifikasi Modul Micro SD Card yang digunakan.

Tabel 4.6 Spesifikasi Modul Micro SD Card

Kategori	Spesifikasi
Tegangan Operasi	3.3V - 5V
Ukuran	20 x 28mm
<i>Interface</i>	SPI (<i>Serial Peripheral Interface</i>)

8. V-Gen Micro SD Card 8gb

V-Gen Micro SD Card merupakan kartu memori penyimpanan yang portable digunakan sebagai tempat menyimpan data gambar yang didapatkan dari modul kamera, data ditulis ke Micro SD Card melalui modul Micro SD Card. Pada penelitian ini menggunakan micro sd card tipe V-Gen. Tabel 4.7 akan menjelaskan tentang spesifikasi dari Micro sd card yang digunakan.

4.7 Tabel Spesifikasi Micro SD Card V-Gen

Kategori	Spesifikasi
Merk	V-Gen
Card Class	10
Dimensi	15 x 11 x 11mm
Ukuran Memori	8 GB
Format	FAT32 / NTFS

Berdasarkan tabel 4.6 micro SD Card V-Gen memiliki besar ruang penyimpanan sebesar 8 giga byte dan *class level* 10. format pada micro SD Card akan dirubah ke dalam format FAT32 karena modul SD hanya dapat membaca kartu dengan format FAT32 dan bukan NTFS.

9. Servo Motor SG90

Servo SG90 Merupakan komponen yang digunakan sebagai aktuator atau penggerak fokus dari Modul Kamera untuk memonitoring dua blok lampu lalu lintas dengan keadaan lampu merah sedang aktif menyala secara bergantian. Servo SG90 memiliki input *analog* yang dapat memberikan hasil output pergerakan sudut dari motor sesuai fokus yang dibutuhkan pada kamera. dan Servo SG90 mampu berputar 180 derajat. Pada tabel 4.8 akan menjelaskan tentang spesifikasi pada Servo SG90.

4.8 Tabel Spesifikasi Servo SG90

Kategori	Spesifikasi
Tegangan Operasi	4.8v ~ 5V
Dimensi	23 x 29 x 12.2mm
Berat	9g (motor)
Material Gear	Nylon
Mode	Analog
Waktu Operasi	0.1 s/60 degree
Range Nilai Operasi	0 - 180 derajat (90 default)

Berdasarkan tabel 4.7 Servo SG90 dapat berputar beroperasi selama 0.1 detik untuk setiap berputaran 60 derajat. Servo SG90 menggunakan input analog untuk mengatur rotasi dibagi menjadi 90 derajat pergerakan ke kiri dan 90 derajat pergerakan ke kanan. Pergerakan servo SG90 aktif digunakan sesuai dengan interval yang akan diberikan pada sistem.

10. BreadBoard

Merupakan komponen papan proyek elektronika yang digunakan untuk membuat rangkaian elektronik dengan menghubungkan kabel atau wiring pada komponen perangkat keras satu ke komponen lainnya. *Breadboard* yang digunakan pada sistem peneliti adalah satu buah *breadboard* dengan tipe ukuran besar yang memiliki jumlah lubang pin sebanyak 830 dan juga dimensi ukuran panjang 175mm dan lebar 67mm dan juga satu buah breadboard berukuran sedang yang memiliki jumlah lubang pin sebanyak 400 dengan dimensi ukuran panjang 85mm dan lebar 67mm.

11. Laptop

Merupakan komponen yang berfungsi untuk sumber daya arduino. Laptop dan juga berfungsi untuk menjalankan aplikasi arduino IDE untuk menulis kode, compile dan upload program ke mikrokontroler serta dapat melakukan *serial monitor* dari eksekusi yang sedang berjalan pada mikrokontroler untuk melakukan pengujian pada sistem. Untuk menjalankan Arduino IDE perangkat PC salah satu sistem operasi yang didukung yang harus terinstall diantaranya Windows 7, Windows 8, Windows 10 maupun Linux. Pada penelitian ini laptop dengan digunakan memiliki spesifikasi yang dijelaskan pada tabel 4.9.

Tabel 4.9 Spesifikasi Laptop yang Digunakan

Kategori	Spesifikasi
Sistem operasi	Windows 7 Ultimate 64 Bit
Ram	8Gb
Prosesor	Intel Core i3-4005U

12. Karton Board Tebal

Pada penelitian ini karton board tebal yang akan dipakai adalah karton memiliki ketebalan 2mm. Karton Board digunakan untuk merancang pembangunan *prototype* sekaligus merupakan komponen yang digunakan sebagai dasar dari peletakan perangkat mikrokontroler dan juga sensor dan juga aktuator yang digunakan pada sistem.

13. Miniatur Mobil

Merupakan miniatur mobil mainan dengan merk Tomica. Miniatur mobil yang digunakan pada penelitian ini memiliki ukuran 6,6cm x 2,8cm dan memiliki perbandingan skala ukuran 1 : 64 dengan mobil asli. komponen miniatur mobil digunakan sebagai ilustrasi pengujian pendeteksian pelanggaran yang terjadi yang akan dilakukan pada prototipe sistem milik peneliti.

4.1.2 Kebutuhan Perangkat Lunak

Kebutuhan antarmuka perangkat lunak merupakan perangkat lunak yang dibutuhkan untuk membangun sistem. Kebutuhan antarmuka perangkat keras akan dijelaskan.

1. Windows 7 Ultimate

Perangkat lunak yang berfungsi sebagai sistem operasi pada perangkat PC atau Laptop yang dipakai dalam penelitian untuk melakukan perancangan sistem yaitu untuk menjalankan Arduino IDE. Juga terdapat beberapa sistem operasi yang didukung diantaranya windows 7, Windows 8, Windows 10 dan juga Linux. Pada penelitian ini peneliti menggunakan Windows 7 sebagai sistem operasi pada perangkat laptop yang digunakan karena dapat menjalankan aplikasi Arduino IDE tanpa adanya kendala.

2. Arduino IDE versi 1.8.4

Merupakan perangkat lunak yang berfungsi sebagai editor untuk menuliskan kode program, *compile* serta *upload* program ke arduino MEGA pada sistem. Bahasa yang digunakan adalah bahasa khusus arduino IDE yang mirip dengan bahasa C. Juga dilengkapi dengan berbagai fungsi dan juga *library* yang mendukung dalam perancangan sistem diantaranya library kamera, servo, dan spi. Arduino IDE juga dapat digunakan sebagai *serial monitoring* untuk melakukan beberapa pengujian pada sistem yang sedang berjalan pada mikrokontroler yang terhubung langsung ke PC. Arduino IDE dapat dioperasikan pada sistem operasi Windows 7, Windows 8, Windows 10 dan juga Linux.

4.2 Kebutuhan *Fungsional*

Kebutuhan *fungsional* merupakan kebutuhan harus dipenuhi agar sistem dapat berjalan sesuai tujuan. Berikut beberapa penjelasan kebutuhan *fungsional* sistem diantaranya :

1. Arudino dapat membaca nilai dari modul LDR untuk mendeteksi gangguan. Fungsi yang menggunakan modul LDR dan *laser diode* pada tiap *block processing* atau tiap blok lampu lalu lintas untuk mendeteksi gangguan pelanggar pada *zebra cross* dengan memanfaatkan konsep *laser security system*. Arduino membaca nilai dari sensor LDR akan mendeteksi gangguan apabila nilai LDR yang dibaca arduino akan menjadi '1' dimana nilai cahaya yang masuk menjadi rendah atau terdapat gangguan pembacaan. Pada saat gangguan terdeteksi maka sistem akan menjalankan fungsi alarm.
2. Timing LED aktif berjalan sesuai dengan code. Fungsi ini mengatur penjadwalan pada LED sebagai indikasi lampu lalu lintas. Terdapat tiga buah LED yaitu merah, kuning dan hijau pada setiap *block processing* yang nilainya berubah ubah tiap kurun waktu tertentu. Lampu merah atau hijau akan aktif selama tiga puluh detik dilanjutkan dengan kedua lampu kuning akan aktif selama tiga detik dan lampu merah dan hijau akan aktif bergantian pada

- setiap lampu lalu lintas sesuai dengan jadwal yang telah ditentukan dengan menggunakan fungsi `Millis()`.
3. Pembacaan pelanggaran aktif sesuai dengan LED yang aktif. Fungsi ini mengatur waktu pendeteksian pelanggaran dengan pembacaan LDR dan *laser diode* harus aktif sesuai lampu merah yang sedang aktif pada blok lampu lalu lintasnya. Jika blok lampu berstatus kuning atau hijau aktif maka fungsi modul LDR dan *laser diode* pada blok tersebut akan non-aktifkan.
 4. Servo sebagai aktuator penggerak fokus kamera. Pada fungsi ini servo digunakan sebagai aktuator untuk menggerakkan fokus pada modul kamera. Fokus pada kamera akan tertuju pada *block processing* dimana lampu lalu lintas dengan LED merah yang sedang aktif berjalan. Penjadwalan dilakukan dengan melakukan `Millis()`, Servo akan aktif menggerakkan fokus kamera pada saat detik ke pertama lampu kuning sedang berjalan.
 5. Buzzer dan kamera aktif sebagai *alarm* pada saat terjadi pelanggaran. Buzzer aktif selama satu detik sebagai *alarm* bunyi dan Kamera akan aktif mengambil gambar apabila terjadi sebuah *trigger* dimana nilai digital modul LDR menjadi '1' atau terdapat objek yang mengganggu pembacaan modul LDR dan *laser diode* dengan kondisi LED merah yang aktif.
 6. Data pengambilan gambar dapat disimpan. Modul Kamera yang telah mengambil gambar dapat menyimpan gambar ke dalam micro sd card. Data dapat disimpan dengan bantuan penulisan data oleh Modul micro sd card.

4.3 Kebutuhan *Non Fungsional*

Kebutuhan *non fungsional* merupakan kebutuhan yang menjelaskan tentang apa saja yang menjadi batasan kebutuhan dalam sistem. Kebutuhan *non fungsional* dari sistem diantaranya sebagai berikut.

4.3.1 Karakteristik Pengguna

Karakteristik pengguna adalah diperuntukan untuk aparat sipil yang menjaga ketertiban serta keamanan pada lalu lintas. Dengan adanya prototipe sistem seperti ini diharapkan dapat memberikan solusi alternatif pada pengguna untuk membangun sistem pengatur lalu lintas serta dapat mengatur ketertiban yang bisa diimplementasikan secara nyata.

4.3.2 Lingkungan Operasi

Prototipe sistem memerlukan kondisi lingkungan baik atau tidak basah maupun lembab oleh faktor cuaca agar sistem dapat bertahan dan pembacaan modul LDR dan juga laser diode dapat berjalan dengan baik.

4.3.3 Asumsi dan Ketergantungan

1. Prototipe Sistem akan aktif berfungsi apabila mikrokontroler telah terhubung ke sumber daya.

2. Letak posisi modul LDR dan *laser diode* harus dalam posisi sejajar berhadapan, dikarenakan modul LDR membutuhkan nilai cahaya *default* dari *laser diode* untuk melakukan pendektasian.
3. Proses penyimpanan data gambar akan dapat dikukan apabila modul SD card telah terpasang micro sd card didalam.

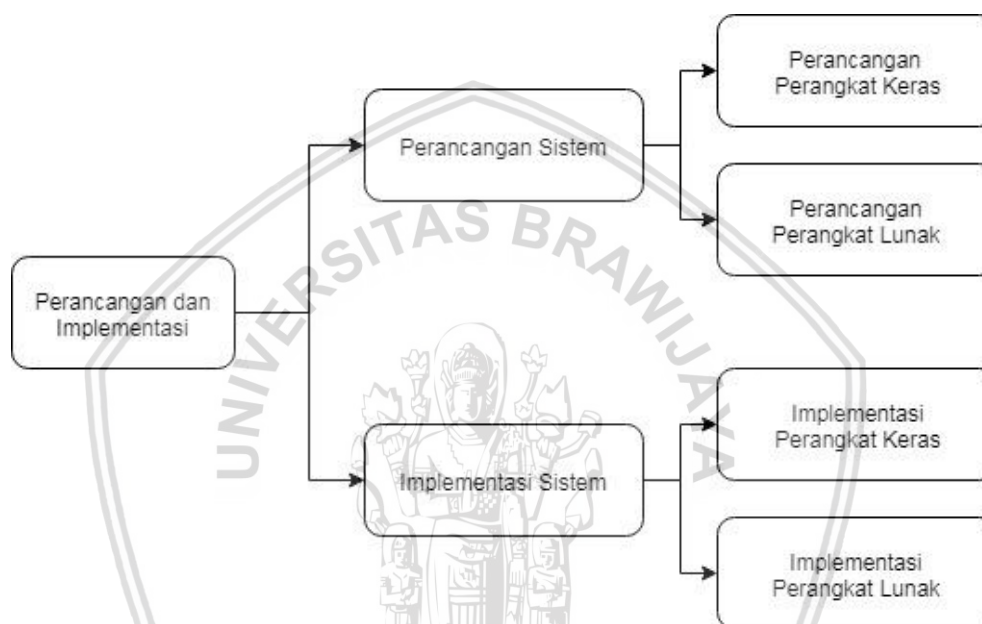
4.3.4 Batasan Perancangan dan Implementasi

Bagian ini akan menjelaskan penentu batasan dari sistem yang dibuat agar penelitian berjalan sesuai harapan. Terdapat batasan sistem yang akan dijelaskan sebagai berikut.

1. Hasil *output alarm* pelanggaran hanya akan aktif sebanyak satu kali pada tiap satu siklus lampu merah yang sedang aktif untuk menghindari adanya *spam*.
2. Penyimpanan data gambar oleh sistem ke dalam micro sd card memiliki batasan jumlah penyimpanan sebesar 8 GB.
3. Sistem dirancang untuk melakukan monitoring hanya dalam ruang lingkup dua persimpangan jalan atau dua lampu lalu lintas.
4. Penggunaan fungsi *Millis()* sebagai pengganti *freeRTOS* untuk mendukung *multitasking* dan *scheduling* pada sistem *real time* pada mikrokontroler arduino dikarenakan *library VC0706* dan *library freeRTOS* tidak saling mendukung satu sama lain.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab perancangan dan implementasi akan menjelaskan tentang perancangan dan implementasi dari sistem prototype pendeteksi pelanggaran pada *zebra cross* menggunakan arduino. Tahapan dilakukan sesuai yang terdapat pada Gambar 5.1. Pertama dilakukan perancangan sistem yang meliputi perancangan perangkat keras dan juga perangkat lunak. Kemudian akan dilakukan implementasi sistem yang meliputi implementasi perangkat keras dan implementasi perangkat lunak. Perancangan dan implementasi dilakukan agar sistem dapat berjalan sesuai dengan tujuan.



Gambar 5.1 Tahapan Perancangan dan Implementasi

Pada perancangan perangkat keras merupakan langkah langkah dalam menghubungkan perangkat-perangkat keras satu sama lain diantaranya hubungan skematik rangkaian antara arduino Mega dengan LED, servo, modul LDR, laser diode, Kamera VC0706, modul SD dan juga buzzer. Kemudian pada perangkat lunak akan menjelaskan gambaran alur pemrograman pada sistem yang akan dirancang.

Setelah tahapan perancangan telah dilakukan, tahapan implementasi sistem akan dilakukan. Pertama merupakan implementasi perangkat keras yang merupakan hasil akhir jadi sistem yang telah dirancang sebelumnya. Kemudian pada implementasi perangkat lunak merupakan hasil source code yang telah dirancang lalu *upload* ke dalam mikrokontroler agar sistem dapat berjalan sesuai yang diinginkan peneliti. *Source code* tersebut memiliki fungsi diantaranya untuk melakukan penjadwalan pada tiap *task*, pembacaan pelanggaran, menghidupkan *alarm*, menjalankan fungsi kamera, serta menyimpan data gambar. Penjelasan lebih lanjut akan dijelaskan pada sub bab-sub bab berikutnya.

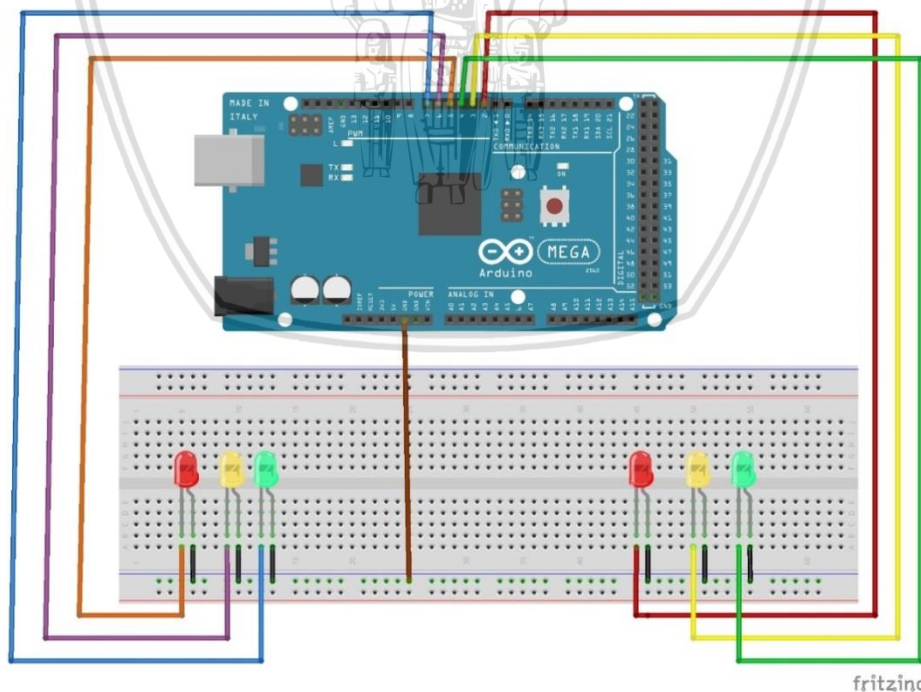
5.1 Perancangan Sistem

Pada sub bab perancangan sistem akan menjelaskan perancangan pada perangkat keras dan juga perangkat lunak. Perancangan perangkat keras meliputi perancangan skematik rangkaian hubungan dari kebutuhan perangkat keras yang digunakan yaitu arduino MEGA dengan beberapa sensor dan aktuator serta penataan tata letak sensor dan aktuator pada prototipe sistem. Perancangan perangkat lunak meliputi penjelasan alur pemograman pada arduino IDE agar sistem dapat berjalan sesuai dengan tujuan pembangunan sistem. Pada sub bab berikutnya akan menjelaskan perancangan pada perangkat keras dan perangkat lunak.

5.1.1 Perancangan Perangkat Keras

Pada sub bab ini akan menjelaskan tentang perancangan perangkat keras yang saling terhubung untuk membangun sistem. Komponen-komponen yang dibutuhkan diantaranya sebuah Arduino Mega yang terhubung dengan Sensor LDR dan *laser diode*, Modul Kamera VC0706, *Traditional Inorganic LED*, Modul Micro SD card, sebuah aktuator Servo SG60, sebuah Buzzer dan perancangan tata letak dan ukuran prototipenya. Setelah analisis pada kebutuhan selanjutnya dilakukan pemasangan komponen perangkat keras serta implementasi. Sub bab berikutnya akan menjelaskan skematik perancangan perangkat keras yang terbagi berdasar tugasnya.

5.1.1.1 Perancangan Skematik Rangkaian LED



Gambar 5.2 Skematik Rangkaian LED

Perancangan yang pertama merupakan perancangan skematik rangkaian LED. Pada Gambar 5.2 merupakan gambaran skematik rangkaian Traditional LED ke

Arduino Mega. *Traditional Inorganic LED* digunakan sebagai simulasi gambaran pada lampu lalu lintas pada umumnya. Terdapat enam buah LED yang digunakan terdiri dari tiga warna yaitu merah, kuning dan hijau sebagai indikasi lampu lalu lintas di setiap bloknya. Tabel 5.1 akan menjelaskan detail dari perangkaian pada skematik rangkaian LED.

Tabel 5.1 Kabel Hubungan LED dengan Arduino

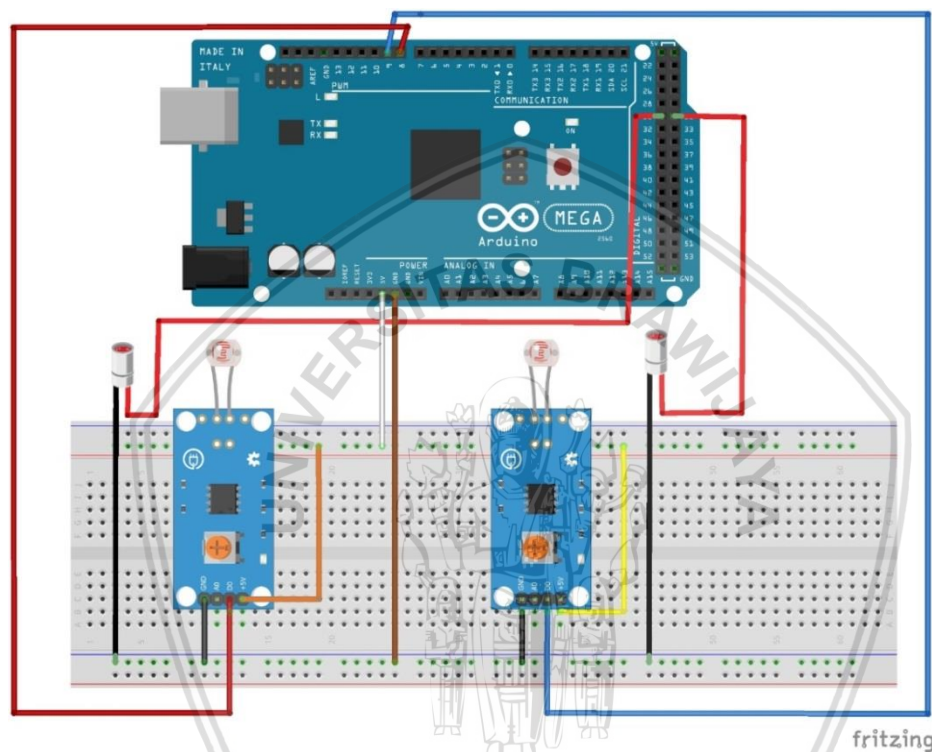
LED Merah 1			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Merah	2	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
LED Kuning 1			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Kuning	3	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
LED Hijau 1			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Hijau	4	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
LED Merah 2			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Orange	5	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
LED Kuning 2			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Ungu	6	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
LED Hijau 2			
Pin LED	Warna Kabel	Pin Arduino	Deskripsi
Out	Biru	7	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>

Tabel 5.1 merupakan konfigurasi LED dengan Arduino Mega, terdapat pin pada tiap LED yang pertama merupakan pin GND yang dihubungkan dengan - atau *Ground* pada *breadboard* yang juga terhubung dengan pin *Ground* pada Arduino. kemudian pada LED terdapat pin output digital pada LED disambungkan ke dalam pin digital pada Arduino yaitu pada pin 2, 3, 4, 5, 6, 7 sehingga kita

dapat mengatur nilai *on/off* atau *HIGH/LOW* pada komponen LED serta mengatur intensitas cahayanya menggunakan fungsi PWM pada Arduino.

5.1.1.2 Perancangan Skematik Rangkaian Modul LDR dan *Laser Diode*

Pada percangan ini akan menjelaskan tentang perancangan skematik raingian modul LDR dan *laser diode*. Modul LDR dan *Laser Diode* digunakan sebagai pendeteksi adanya gangguan atau pelanggaran yang terjadi di dalam suatu blok lampu lalu lintas. Pada Gambar 5.3 merupakan gambaran skematik rangkaian Modul LDR dan *Laser Diode* ke Arduino Mega:



Gambar 5.3 Skematik Rangkaian LDR dan *Laser Diode*

Tabel 5.2 akan menjelaskan detail dari skematik rangkaian LDR dan *laser diode* pada Gambar 5.3:

Tabel 5.2 Kabel Hubungan LDR dan *Laser Diode* dengan Arduino

Pin LDR 1	Warna Kabel	Pin Arduino	Deskripsi
DO	Biru	9	<i>Digital Output</i>
+5V	Kuning	5V	<i>5v DC Power</i>
GND	Hitam	GND	<i>Ground</i>
Pin Laser 1	Warna Kabel	Pin Arduino	Deskripsi
Out	Merah	31	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
Pin LDR 2	Warna Kabel	Pin Arduino	Deskripsi

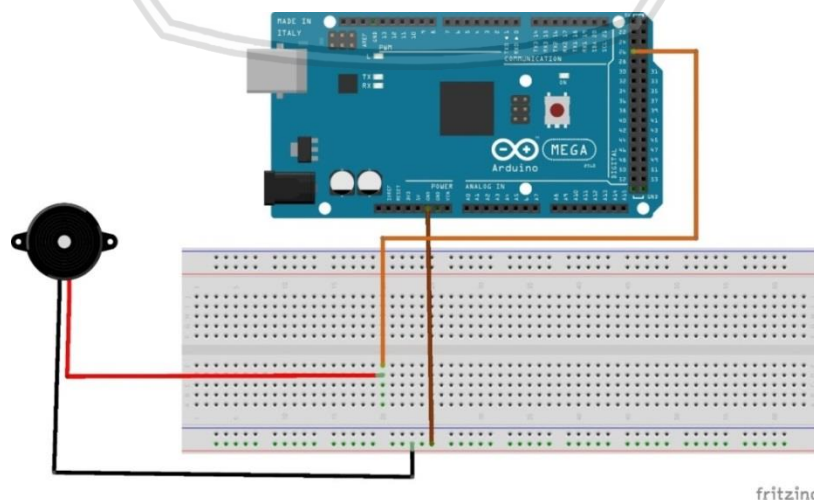
DO	Merah	8	Digital Output
+5V	Orange	5V	5v DC Power
GND	Hitam	GND	Ground
Pin Laser 2	Warna Kabel	Pin Arduino	Deskripsi
Out	Merah	30	Output
GND	Hitam	GND	Ground

Pada Tabel 5.2 merupakan konfigurasi Modul LDR dan *laser diode* dengan Arduino Mega, terdapat 4 pin yang digunakan pada tiap modul LDR , yaitu pin +5V dihubungkan dengan + atau Vcc pada *breadboard* yang terhubung dengan 5V pada Arduino, pin GND yang dihubungkan ke - atau *Ground* pada *breadboard* yang terhubung dengan pin *Ground* pada Arduino, Serta pin DO yang merupakan pin untuk konfigurasi *digital output* pada LDR dihubungkan dengan pin digital PWM 8 dan 9 pada Arduino Mega. Modul LDR mampu menghasilkan *analog* dan *digital output*, untuk penelitian ini akan memanfaatkan *digital output* pada LDR.

Pada konfigurasi *laser diode* terdapat dua kabel merah dan juga hitam. Kabel Hitam disambungkan ke - atau *Ground* pada *breadboard* yang juga terhubung dengan *Ground* pada Arduino. Kabel merah merupakan digital output dari *laser diode* yang berfungsi untuk mengatur nilai *on/off* disambungkan ke pin digital 30 dan 31 pada Arduino Mega. Sehingga komponen-komponen yang diperlukan fungsi pendeteksi dengan menggunakan konsep *laser security* pada sistem dapat berjalan.

5.1.1.3 Perancangan Skematik Rangkaian Buzzer

Pada perancangan berikutnya merupakan perancangan skematik rangkaian pada buzzer. Buzzer digunakan sebagai *alarm* pada saat terjadinya sebuah pelanggaran sedang terjadi. Buzzer menghasilkan output berupa getaran dan bunyi. Pada Gambar 5.4 merupakan gambaran skematik rangkaian Buzzer ke Arduino Mega:



Gambar 5.4 Skematik Rangkaian Buzzer

Tabel 5.3 akan menjelaskan detail dari perangkaian pada skematik rangkaian Buzzer pada Gambar 5.4:

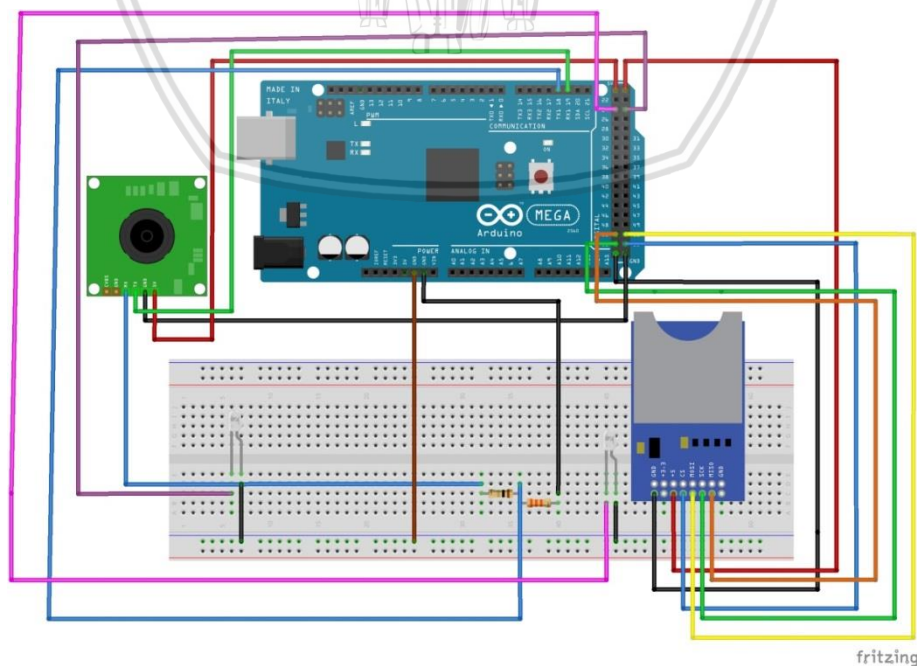
Tabel 5.3 Kabel Hubungan Buzzer dengan Arduino

Pin Buzzer	Warna Kabel	Pin Arduino	Deskripsi
Out	Merah	26	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>

Tabel 5.3 merupakan konfigurasi Buzzer dengan Arduino Mega, pada buzzer terdapat kabel hitam yang merupakan GND yang dihubungkan dengan - atau *Ground* pada *breadboard* yang juga terhubung dengan Ground pada Arduino. kemudian pada kabel merah merupakan *digital output* pada Buzzer disambungkan ke dalam pin digital pada Arduino yaitu pada pin 26. sehingga sistem dapat mengatur nilai *on/off* atau *HIGH/LOW* pada komponen. Buzzer juga dapat mengatur frekuensi dan durasi pada saat aktif dengan menggunakan fungsi *Tone* pada Arduino IDE.

5.1.1.4 Perancangan Skematik Rangkaian VC0706 Dan Modul Micro SD

Perancangan selanjutnya adalah perancangan skematik rangkaian VC0706 dan Modul Micro SD. Modul Kamera VC0706 berfungsi untuk mengambil sebuah gambar pada saat terdeteksi sebuah *trigger* pelanggaran sedangkan Modul Micro SD Card mini digunakan sebagai penulisan data gambar yang telah diambil oleh modul kamera menuju micro sd card. Pada Kedua Modul dilengkapi juga dengan sebuah LED sebagai indikasi terjadinya gangguan inisialisasi pada modul Kamera dan juga Modul SD Card. Pada Gambar 5.5 merupakan gambaran skematik rangkaian Modul Kamera VC0706 Dan Modul Micro SD Card TF mini ke Arduino Mega:



Gambar 5.5 Skematik Rangkaian VC0706 Dan Modul SD

Tabel 5.4 akan menjelaskan detail dari perangkain pada skematik rangkain VC0706 dan Modul SD pada Gambar 5.5:

Tabel 5.4 Kabel Hubungan VC0706 dan Modul SD dengan Arduino

Pin VC0706	Warna Kabel	Pin Arduino	Deskripsi
TX	Hijau	19 RX	<i>Receive</i>
RX	Biru	18 TX	<i>Transmit</i>
GND	Hitam	GND	<i>Ground</i>
+5V	Merah	5V	<i>5v DC Power</i>
Pin Modul Micro SD Card	Warna Kabel	Pin Arduino	Deskripsi
CS	Biru	53	<i>Card Select</i>
SCK	Hijau	52	<i>Serial Clock</i>
MOSI	Kuning	51	<i>Data Input</i>
MISO	Orange	50	<i>Data Output</i>
VCC	Merah	5V	<i>5v DC Power</i>
GND	Hitam	GND	<i>Ground</i>
Pin LED Kamera	Warna Kabel	Pin Arduino	Deskripsi
OUT	Ungu	25	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>
Pin LED SD Card	Warna Kabel	Pin Arduino	Deskripsi
OUT	Merah muda	24	<i>Output</i>
GND	Hitam	GND	<i>Ground</i>

Pada Tabel 5.4 merupakan konfigurasi Modul Kamera VC0706, Modul Micro SD Card dan LED dengan Arduino Mega. Pada VC0706 terdapat 4 pin yang akan digunakan yaitu pin +5V dihubungkan dengan + atau Vcc pada *breadboard* yang terhubung dengan pin 5V pada Arduino, pin GND yang dihubungkan ke - atau *Ground* pada *breadboard* yang terhubung dengan pin *Ground* pada Arduino, lalu terdapat pin TX yang disambungkan dengan pin 19 RX pada Arduino dan pin RX yang terhubung dengan pin 18 TX. Pada Gambar 5.3 terdapat resistor yang membagi tegangan pada kabel RX pada modul kamera, karena logic level pada komunikasi serial kamera data pin adalah 3.3v sedangkan logic level pada Arduino Mega adalah 5V. RX berfungsi sebagai jalur penerimaan data dari modul ke komponen lain, sedangkan TX berfungsi untuk mengirim data atau jalur pengiriman data antara modul ke komponen lain. Data yang dikirimkan pada modul VC0706 dapat berupa video maupun *snapshot* gambar.

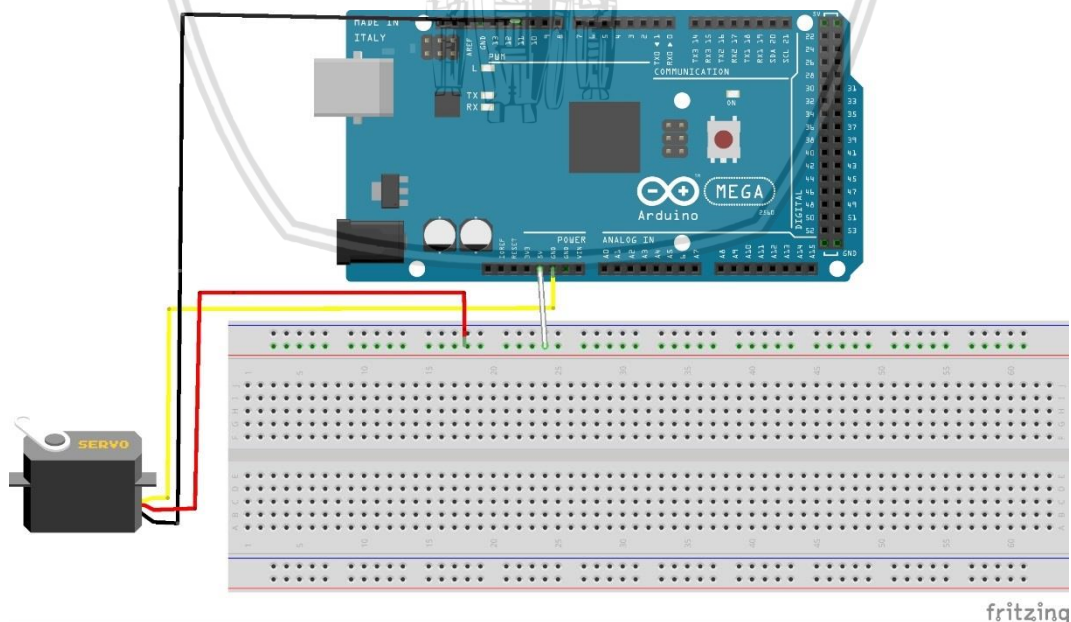
Pada konfigurasi Modul micro SD terdapat beberapa pin yang dihubungkan. Pada pin GND disambungkan ke - atau *Ground* pada *breadboard* yang juga

terhubung dengan pin *Ground* pada Arduino. Pada pin *+VCC* dihubungkan dengan *+* atau *Vcc* pada *breadboard* yang terhubung dengan pin 5V. selanjutnya pada pin CS merupakan Card Select yaitu pin yang digunakan untuk mengaktifkan pembacaan micro SD Card yang terpasang pada modul, disambungkan ke pin digital 53 pada Arduino Mega. Kemudian terdapat pin MOSI (Master Out Slave In) dan MISO (Master In Slave Out) yang terhubung pada pin digital 51 dan 50 pada arduino Mega, MISO berfungsi sebagai jalur yang digunakan untuk menerima sedangkan MOSI adalah jalur yang digunakan untuk mengirimkan data ke mikrokontroler. Pada pin SCK atau disebut *Serial Clock* dihubungkan dengan pin 52 pada Arduino, fungsi pin SCK adalah sebagai jalur komunikasi yang membantu sinkronisasi antara MISO dan MOSI.

Terdapat dua buah LED yang digunakan sebagai indikasi kamera sedang aktif berjalan dan indikasi error pada pembacaan micro SD card. Pada konfigurasi LED terdapat pin pada tiap LED yang pertama merupakan pin GND yang dihubungkan dengan *-* atau *Ground* pada *breadboard* yang juga terhubung dengan pin Ground pada Arduino. kemudian pada LED terdapat pin output digital pada LED disambungkan ke dalam pin digital pada Arduino yaitu pada pin 24, 25 sehingga sistem dapat mengatur nilai *on/off* atau *HIGH/LOW* pada LED sesuai dengan tujuannya.

5.1.1.5 Perancangan Skematik Rangkaian Aktuator Servo SG90

Servo SG90 digunakan sebagai penggerak kamera untuk menentukan fokus kamera menuju blok lampu lalu lintas dimana lampu merah sedang aktif. Pada Gambar 5.6 merupakan gambaran skematik rangkaian servo SG90 ke Arduino Mega:



Gambar 5.6 Skematik Rangkaian SG90

Tabel 5.5 akan menjelaskan detail dari perangkaian pada skematik rangkaian servo SG90 pada Gambar 5.6:

Tabel 5.5 Kabel Hubungan SG90 dengan Arduino

Pin SG90	Warna Kabel	Pin Arduino	Deskripsi
VCC	Merah	5V	<i>5v DC power</i>
GND	Hitam	GND	<i>Ground</i>
Out	Kuning	11	<i>Output</i>

Pada Tabel 5.5 merupakan konfigurasi Aktuator servo SG90 dengan Arduino Mega, terdapat 3 pin yang digunakan pada SG90, yaitu pin +5V dihubungkan dengan + atau Vcc pada *breadboard* yang terhubung dengan pin 5V pada Arduino, pin GND yang dihubungkan ke - atau *Ground* pada *breadboard* yang terhubung dengan pin *Ground* pada Arduino, Serta pin Output yang merupakan pin untuk konfigurasi *output* pada SG90 dihubungkan dengan pin digital PWM 11 pada Arduino Mega.

Output pada SG90 berupa nilai *analog* yang berfungsi untuk mengatur sudut perputaran aktuator dengan nilai maksimal sampai 180 derajat, yang dibagi menjadi 90 derajat ke arah kiri dan juga kanan.

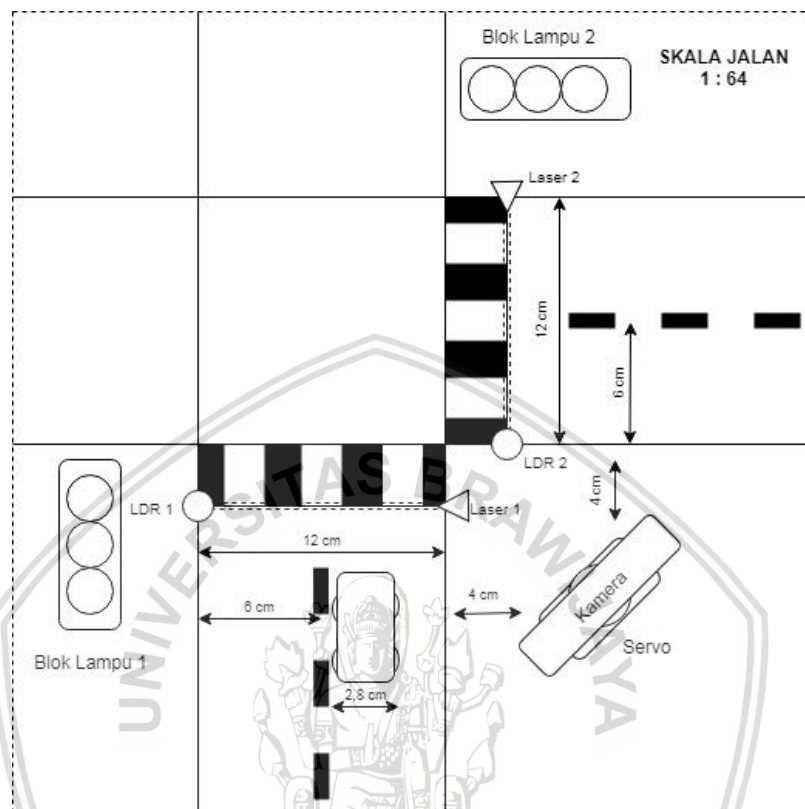
5.1.1.6 Perancangan Tata Letak dan Ukuran Prototipe

Pada perancangan ini merupakan gambaran tentang tata letak pada perancangan perangkat keras dan pengukuran pada prototipe sistem peneliti. Ukuran jalur pada prototipe yang akan dibangun ini akan menyesuaikan dengan skala pada model miniatur kendaraan mobil yang digunakan yaitu memiliki perbandingan ukuran skala 1 : 64 dengan ukuran yang nyata. Perancangan ini difokuskan pada pengukuran jalur lalu lintas yang pada *prototype* sekaligus tata letak posisi sensor pendeteksi pada LDR dan laser diode dan tata letak servo sebagai penumpu letak pada modul kamera yang digunakan untuk mendapatkan data gambar pada jarak yang optimal yang difokuskan untuk memantau zebra cross pada blok lampu 1 dan blok lampu 2.

Berdasarkan Departemen Pekerjaan Umum (1997) lebar tiap lajur lalu-lintas efektif pada umumnya memiliki lebar 3 meter hingga 4 meter. Pada prototipe sistem peneliti terdapat dua persimpangan jalan atau dua jalur yang memiliki *zebra cross* sekaligus lampu lalu lintas yaitu blok lampu 1 dan 2 disetiap jalurnya dan disetiap jalurnya masing-masing terdiri dari dua lajur satu arah. Berdasarkan ukuran skala 1 : 64 yang telah ditentukan sekaligus berdasarkan lebar lajur jalan efektif yang disarankan, pada *prototype* sistem peneliti menggunakan lebar jalur masing masing 12 cm atau 6 cm disetiap jalurnya yang mewakili ukuran nyata yaitu jalur jalan selebar 7,6 meter dan lajur jalan selebar 3,8 meter.

Pengukuran lebar jalan yang telah dilakukan juga merupakan jarak lebar deteksi pembacaan pelanggar oleh modul LDR dan juga laser yang terpasang pada prototipe. Terdapat dua pasang LDR dan *laser diode* yang terpasang pada

posisi bersebrangan pada tiap jalur, posisi modul LDR disejajarkan sedang posisi *laser diode* agar *laser diode* dapat memberika nilai cahaya *default* yang akurat. Berikut Gambar 5.7 merupakan tata letak sekaligus pengukuran *prototype* yang akan diimplementasikan pada penelitian ini.



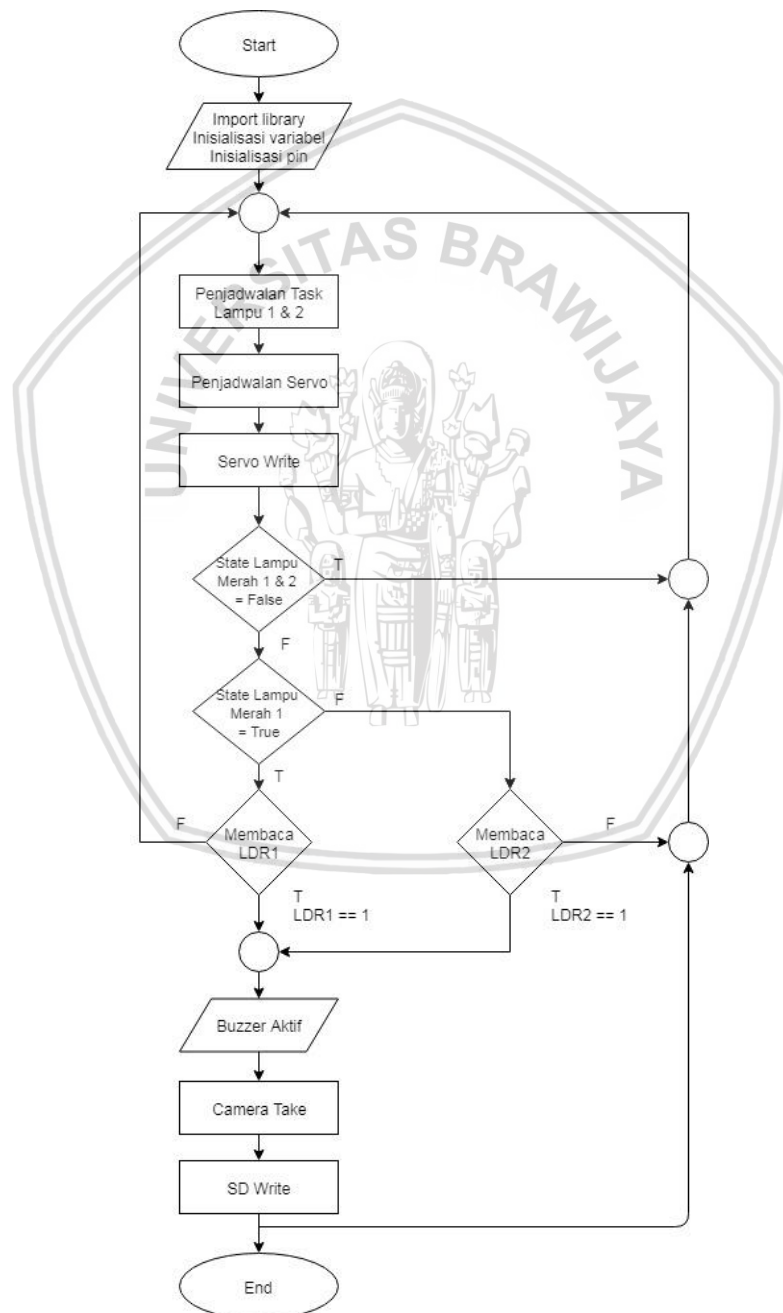
Gambar 5.7 Perancangan Tata Letak dan Ukuran Prototipe

Pada Gambar 5.7 servo dihubungkan dengan modul kamera yang nantinya digunakan sebagai penggerak fokus pada kamera, secara bergantian menggerakkan posisi sudut kamera menuju blok lampu 1 dan menggerakkan posisi sudut kamera menuju blok lampu 2. Pada peletakannya, servo dan kamera diletakan pada tepian luar jalur dengan masing-masing jarak sejauh 4 cm agar kamera yang terpasang pada servo dapat melakukan pengambilan gambar yang optimal pada objek yang melanggar dengan menggunakan resolusi 640x480 pixel maupun 320x240 pixel. Lalu berdasarkan posisi peletakannya, kamera akan mendapatkan fokus yang optimal dengan memutar sudut 20 sampai 30 derajat menuju *zebra cross* pada blok lampu yang aktif. Perancangan akan dilakukan pada kertas karton board tebal dengan ukuran panjang 38 cm dan lebar 32 cm.

5.1.2 Perancangan Perangkat Lunak

Pada perancangan perangkat lunak dalam pembangunan sistem ini meliputi gambaran tentang alur kerja dari perangkat lunak yang akan dibangun dengan menggunakan Arduino IDE. Pertama melakukan import beberapa library yang dibutuhkan serta inisialisasi pin-pin dan *variable*, lalu pemanggilan *task* penjadwalannya lampu lalu lintas satu dan dua sebagai indikasi status pada tiap

blok lampu lalu lintas dan juga pergerakan servo. Dalam studi kasus penelitian ini menggunakan dua persimpangan jalan. Serta pembacaan nilai pada LDR sebagai pendeteksi adanya pelanggaran disuatu blok lampu pada saat keadaan lampu atau LED merahnya sedang aktif. Pada saat sebuah pelanggaran terdeteksi atau pembacaan nilai LDR menjadi '1' pada suatu blok lampu, maka *alarm* akan aktif dan kamera akan mengambil gambar. Hasil gambar yang didapatkan dari modul kamera akan dituliskan ke dalam micro sd card yang tersimpan didalam modul micro sd. Perancangan perangkat lunak dituliskan pada aplikasi Arduino IDE dan *script* akan diupload ke dalam mikrokontroler Arduino Mega. Gambar 5.8 dibawah ini merupakan diagram alir dari perancangan perangkat lunak:



Gambar 5.8 Diagram Alir Perancangan Perangkat Lunak

Berikut ini merupakan penjelasan dari diagram alir peranan perangkat lunak pada Gambar 5.8:

1. Pertama-tama perancangan akan diawali dengan *import* beberapa *library* serta inisialisasi nilai awal dari beberapa *variable* dan juga pengaturan pin-pin yang terhubung.
2. Kemudian sistem akan melakukan penjadwalan LED yang aktif bergantian disetiap blok serta menjalankan servo secara berkala menggunakan fungsi *millis()*.
3. Pembacaan LDR dan *laser diode* akan aktif pada salah satu blok apabila lampu ketika LED merah sedang aktif. Pembacaan pada LDR tidak akan aktif ketika kedua lampu kuning sedang menyala. sistem akan membaca nilai dari LDR untuk mendeteksi adanya gangguan pembacaan nilai.
4. Ketika nilai LDR yang berubah karena adanya gangguan yang terjadi, maka sistem akan mengaktifkan buzzer sebagai *alarm* serta menjalankan kamera untuk mengambil gambar.
5. Gambar yang telah diambil akan dituliskan lalu disimpan pada micro sd card melalui modul micro sd card.
6. Sistem akan terus melakukan perulangan kembali ke penjadwalan LED dan servo serta pendeteksian pada pelanggar.

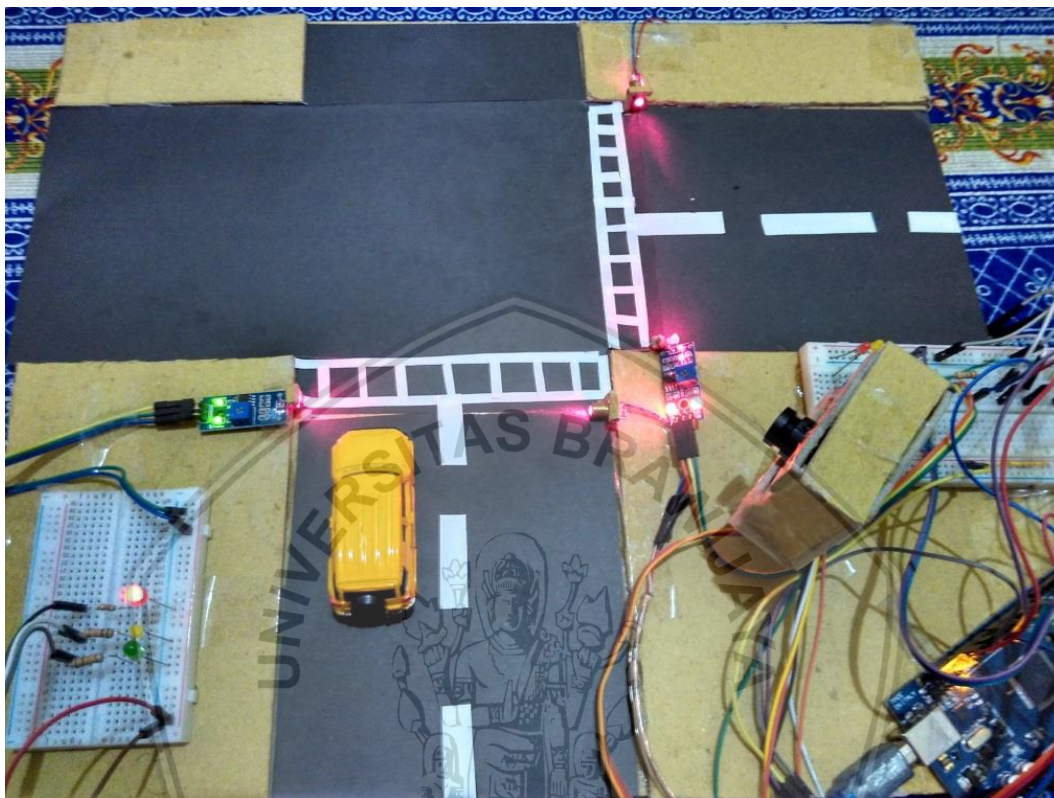
5.2 Implementasi Sistem

Pada Sub bab ini akan dilakukan ketika proses perancangan telah selesai, implementasi dilakukan sesuai dengan perancangan yang dibuat pada sub bab sebelumnya agar sistem dapat berjalan sesuai dengan tujuan yang diharapkan. Implementasi sistem akan menjelaskan secara rinci mengenai implementasi perangkat keras yang telah dirancang yaitu tata letak perancangan pada prototipe sistem dan implementasi perangkat lunak yang berupa penjelasan *source code* yang telah dirancang.

5.2.1 Implementasi Perangkat Keras

Pada implementasi perangkat keras merupakan hasil implementasi dari perancangan perangkat keras yang telah dilakukan sebelumnya baik dalam perancangan skematik dan juga tata letak dari perangkat-perangkat keras. Pada Hasil implementasi yang dilakukan, prototipe sistem terdiri dari mikrokontroler arduino MEGA digunakan sebagai pusat kontrol untuk mengatur sistem. Terdapat beberapa komponen perangkat keras yang telah terhubung diantaranya delapan buah *Traditional Inorganic LED*, enam buah sebagai indikasi lampu lalu lintas, dua pasang LED untuk indikasi status aktif pada modul kamera dan modul SD yang telah terhubung dengan micro SD card, serta terdapat dua pasang modul LDR dan *laser diode* sebagai sensor pendeteksi adanya pelanggaran pada tiap blok lampu lalu lintas, satu buah Buzzer sebagai *alarm* suara, satu buah

Kamera VC0706 serta modul micro sd card SPI yang telah terpasang micro SD Card untuk melakukan penyimpanan data pelanggar dan satu buah aktuator servo SG90. Pada Gambar 5.9 merupakan hasil implementasi dari perancangan perangkat keras yang telah dilakukan, gambar ditampilkan pada tampak atas dari prototipe sistem yang diimplementasikan.



Gambar 5.9 Tampak Atas Perancangan Perangkat Keras

Pada Gambar 5.9 menampilkan tampak atas dari rancangan perangkat keras yang telah dibangun, dimana terdapat sebuah mikrokontroler Arduino Mega, yang terhubung dengan dua *breadboard* dan terdapat komponen perangkat yang terhubung delapan buah LED, satu *Buzzer*, dua *LDR Module* dan dua *laser diode*, satu buah modul *VC0706 Camera*, satu *Micro SD Card SPI TF module* dan *servo SG90*. Komponen yang digunakan untuk melakukan penjadwalan pada simulasi lampu lalu lintas adalah lampu LED merah, kuning dan hijau disetiap blok lampu lalu lintas. Serta Aktuator servo yang disambungkan dengan Modul Kamera yang berfungsi untuk mengatur perpindahan fokus kamera dari blok lampu satu ke blok lampu merah lainnya, dimana kondisi lampu merah akan aktif menyala. Penjadwalan akan diatur dengan menggunakan fungsi *Millis*, sehingga LED dan servo mampu aktif dalam kurun waktu yang telah ditentukan.

Komponen pendeteksi yang memanfaatkan konsep *laser security system* diimplementasikan dengan memasang komponen LDR dan juga *laser diode* dalam keadaan sejajar sehingga *laser diode* dapat memberikan nilai *default* LDR yang pasti. Pendeteksian dilakukan dengan pembacaan nilai dari modul LDR oleh Arduino Mega ketika status LED merah sedang aktif., Pembacaan nilai tersebut

terganggu apabila terdapat sebuah objek yang menghalangi pembacaan LDR dengan *laser diode*. Buzzer dan VC0706 akan aktif bekerja ketika terdeteksi adanya sebuah pelanggaran. Lalu data gambar yang didapatkan oleh VC0706 akan dituliskan ke dalam micro sd card yang terhubung dalam modul Micro SD Card. Pada hasil implementasi perangkat keras sistem komponen-komponen disusun pada kertas karton tebal

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak dalam sistem ini adalah penulisan kode pada Arduino IDE untuk mikrokontroler Arduino Mega. Penulisan kode pada Arduino IDE akan dilakukan didalam laptop peneliti pada operating sistem Windows7. Kode yang telah ditulis pada Arduino IDE akan di upload ke dalam mikrokontroler Arduino Mega agar sistem dapat berjalan sesuai dengan tujuan yang dibangun. Terdapat beberapa fungsi yang mengatur pekerjaan dari komponen modul sensor maupun aktuator. Penjelasan *source code* sistem akan dijelaskan secara rinci sebagai berikut

5.2.2.1 Implementasi Setup Sistem

Implementasi perangkat lunak yang pertama adalah melakukan pengaturan atau konfigurasi yang diperlukan pada sistem. Pada potongan *Source code* yang pertama yaitu pada Kode Program 5.1 digunakan untuk mengimport beberapa *library* pendukung yang dibutuhkan agar sistem dapat mengatur beberapa fungsi serta pembacaan sensor dan aktuator yang terhubung dengan mikrokontroler Arduino Mega. Terdapat beberapa *library* yang dibutuhkan yaitu library kamera VC0706, serta komunikasi SPI, SD modul serta servo.

Kode Program 5.1 Import Library

	Source Code
1	<code>#include <Adafruit_VC0706.h></code>
2	<code>#include <SPI.h></code>
3	<code>#include <SdFat.h></code>
4	<code>#include <Servo.h></code>
5	<code>Adafruit_VC0706 cam = Adafruit_VC0706(&Serial1);</code>

Lalu dilakukan pengaturan insialisasi nilai awal dari Millis yaitu nilai waktu terakhir pengerjaan *task* pada Servo dan juga Lampu. Serta inisialisasi variabel *interval* waktu tiap pengerjaan *task*nya sebesar 1000 ms atau 1 detik seperti yang terlihat pada Kode Program 5.2.

Kode Program 5.2 Pengaturan Millis

	Source Code
1	<code>unsigned long previousMillisServo = 0;</code>
2	<code>unsigned long previousMillisLampu = 0;</code>

3	<code>const long intervaltask = 1000;</code>
---	--

Kemudian dilakukan setup awal insialisasi pin yang terhubung pada serta melakukan setting *baud rate* maksimal sebesar 115200 yang merupakan konfigurasi kecepatan pembacaan data bit perdetik pada komunikasi serial pada kamera. Berfungsi untuk membantu sistem untuk meningkatkan performansi penulisan data gambar ke dalam micro sd card sesuai yang terlihat pada Kode Program 5.3.

Kode Program 5.3 Pengaturan Pin

	Source Code
1	<code>Serial.begin(115200);</code>
2	<code>#define SD_CS 53</code>
3	<code>const int laser1 = 12;</code>
4	<code>const int laser2 = 13;</code>
5	<code>const int lampucam = 24;</code>
6	<code>const int lampucamera = 25;</code>
7	<code>const int buzzer = 31;</code>
8	<code>int detiklampu = 0;</code>
9	<code>for (int i = 0; i < 3; i++) {</code>
10	<code>pinMode(lampu1[i], OUTPUT);</code>
11	<code>pinMode(lampu2[i], OUTPUT);</code>
12	<code>}</code>
13	<code>myservo.attach(11);</code>
14	<code>pinMode(lampumerah1, INPUT);</code>
15	<code>pinMode(lampumerah2, INPUT);</code>
16	<code>pinMode(lampucamera, INPUT);</code>
17	<code>pinMode(lampucam, INPUT);</code>
18	<code>pinMode(buzzer, OUTPUT);</code>
19	<code>pinMode(laser1, OUTPUT);</code>
20	<code>pinMode(laser2, OUTPUT);</code>
21	<code>SdFat SD;</code>
22	<code>Adafruit_VC0706 cam = Adafruit_VC0706(&Serial1);</code>
23	<code>Servo myservo;</code>

Berdasarkan Kode Program 5.3 melakukan konfigurasi pin-pin yang terhubung pada perancangan perangkat keras diantara pin pada LED, SD, servo, Kamera, Buzzer, LDR dan juga *laser diode*.

Selanjutnya akan dilakukan inialisasi nilai awal dari beberapa variable yang dibutuhkan pada sistem diantaranya adalah lampu state yang berfungsi sebagai nilai akses perijinan pembacaan pada LDR untuk mendeteksi pelanggaran sesuai dengan kondisi lampu yang sedang aktif, lalu inialisasi variable pembatasan

pengaktifan *alarm* agar tidak terjadi *spam*. Serta nilai deti lampu dan detikservo yang digunakan untuk perhitungan waktu detik yang berjalan pada saat *task* akan dieksekusi. Dan melakukan mengaktifkan laser diode 1 dan 2 untuk memberi nilai cahaya default ke sensor LDR yang telah terpasang serta inisialisasi posisi awal servo yang mengatur fokus kamera menuju blok lampu merah 1 yang akan aktif pada saat sistem di jalan sesuai pada Kode Program 5.4.

Kode Program 5.4 Pengaturan Variabel

	Source Code
1	<code>bool lampustate1 = false;</code>
2	<code>bool lampustate2 = false;</code>
3	<code>bool donetake = false;</code>
4	<code>int detiklampu = 0;</code>
5	<code>int detikservo = 0;</code>
6	<code>digitalWrite(laser1, HIGH);</code>
7	<code>digitalWrite(laser2, HIGH);</code>
8	<code>myservo.write(105);</code>

5.2.2.2 Implementasi Main Function

Merupakan fungsi utama pada program berupa *loop* atau perulangan pada program utama yang terus aktif berjalan. Guna untuk menjalankan fungsi-fungsi sesuai dengan jadwalnya diantaranya fungsi pengaktifan deteksi, lampu, servo serta fungsi deteksi. Pada kode program 5.5 merupakan *source code* yang digunakan untuk mengukur waktu yang telah berjalan pada sistem dan menyimpannya pada variabel menggunakan fungsi `Millis()`.

Kode Program 5.5 Pengukuran waktu yang berjalan pada sistem

	Source Code
1	<code>void loop() {</code>
2	<code> unsigned long currentMillis = millis();</code>

Source berikutnya merupakan pembacaan nilai pada dua sensor LDR dilakukan berdasarkan kondisi lampu state merah yang sedang aktif, lalu dilakukan pemanggilan fungsi `deteksi()` yang berfungsi untuk pengecekan kondisi pelanggaran pada saat terdeteksi sesuai pada Kode 5.6.

Kode Program 5.6 Pembacaan nilai LDR

	Source Code
1	<code>if (lampustate1) {</code>
2	<code> LDR1 = digitalRead(lampumerah1);</code>
3	<code>}</code>

4	else if (lampustate2) {
5	LDR2 = digitalRead(lampumerah2);
6	}
7	Deteksi();

Source berikutnya merupakan pemanggilan fungsi task lampu dengan memanfaatkan fungsi millis(). Task dieksekusi berdasarkan nilai waktu yang berjalan dikurangi waktu task yang telah berjalan sebelumnya dibandingkan dengan interval waktu task akan dikerjakan yang telah di set sebelumnya sebesar 1000 ms atau 1 detik. Task akan berjalan memanggil fungsi lampu setiap 1 detik berjalan sesuai Kode Program 5.7.

Kode Program 5.7 Pemanggilan Fungsi Lampu dengan interval 1 detik

	Source Code
1	if (currentMillis - previousMillisLampu >=
2	intervallampumerah) {
3	previousMillisLampu = currentMillis;
4	fungsiilampumerah();
5	}
6	}

Berikutnya merupakan pemanggilan fungsi task servo dengan memanfaatkan fungsi millis(). Task dieksekusi berdasarkan nilai waktu yang berjalan dikurangi waktu task yang telah berjalan sebelumnya dibandingkan dengan interval waktu task akan dikerjakan yang telah di set sebelumnya sebesar 1000 ms atau 1 detik. Task akan berjalan memanggil fungsi servo setiap 1 detik berjalan sesuai Kode Program 5.8.

Kode Program 5.8 Pemanggilan Fungsi Servo dengan interval 1 detik

	Source Code
1	if (currentMillis - previousMillisServo >=
2	intervallampumerah) {
3	previousMillisServo = currentMillis;
4	fungsiiservo();
5	}

5.2.2.3 Implementasi Fungsi Lampu

Source code ini merupakan fungsi yang dipanggil pada millis dengan *interval* waktu satu detik untuk mengatur nyala lampu pada LED sebagai indikasi lampu lalu lintas 1 dan 2. Nyala lampu merah dan hijau akan aktif selama 30 detik bergantian dengan selang waktu 3 detik waktu lampu kuning akan aktif. Fungsi pembacaan pelanggaran pada sensor LDR akan diaktifkan berdasarkan lampu merah yang aktif. nilai perulangan akan kembali ke awal apabila telah mencapai nilai detik ke 66 atau lampu kuning ke dua berakhir. Nilai pengijinan untuk

melakukan eksekusi pelanggar berikutnya juga akan diaktifkan kembali pada saat loop telah direset pada detik ke 6.

Berikut merupakan *Source code* yang digunakan untuk mengatur penjadwalan lampu pada 30 detik pertama yaitu nilai LED merah pada blok 1 dan nilai LED hijau pada blok 2 akan aktif bersamaan. Serta menyalakan pembacaan deteksi sensor LDR pada blok lampu 1 sesuai dengan kondisi lampu merah yang sedang aktif pada sesuai dengan Kode Program 5.9.

Kode 5.9 Fungsi Lampu mengatur 30 detik awal

	Source Code
1	void fungsilampumerah() {
2	//lampu merah1 akan aktif selama 30detik
3	//lampu hijau2 akan aktif selama 30detik
4	if (detiklampu1 >= 0 && detiklampu <30) {
5	detiklampu += 1;
6	if (detiklampu == 1) {
7	Serial.println("Lampu1 sedang merah!");
8	Serial.println("Lampu2 sedang hijau!");
9	}
10	Serial.println(detiklampu1);
11	digitalWrite(lampu1[2], LOW);
12	digitalWrite(lampu1[1], LOW);
13	digitalWrite(lampu1[0], HIGH);
14	digitalWrite(lampu2[0], LOW);
15	digitalWrite(lampu2[1], LOW);
16	digitalWrite(lampu2[2], HIGH);
17	lampustate1 = true;
18	lampustate2 = false;
19	}

Selanjutnya merupakan *Source code* yang digunakan untuk mengatur penjadwalan lampu pada detik ke 30 hingga detik ke 33 yaitu mengaktifkan nilai LED kuning pada blok 1 dan juga nilai LED kuning pada blok 2 serta akan mematikan fungsi pembacaan LDR pada berdasarkan kondisi kedua lampu berada pada kondisi kuning sedang aktif menyala sesuai Kode Program 5.10.

Kode Program 5.10 Fungsi Lampu mengatur detik ke 30 hingga 33

	Source Code
1	//lampu kuning 1&2 akan aktif selama 3detik
2	else if (detiklampu >= 30 && detiklampu <33) {

3	detiklampu += 1;
4	if (detiklampu == 31) {
5	Serial.println("Lampu1 sedang kuning!");
6	Serial.println("Lampu2 sedang kuning!");
7	}
8	Serial.println(detiklampu-30);
9	digitalWrite(lampu1[0], LOW);
10	digitalWrite(lampu1[1], HIGH);
11	digitalWrite(lampu1[2], LOW);
12	digitalWrite(lampu2[0], LOW);
13	digitalWrite(lampu2[1], HIGH);
14	digitalWrite(lampu2[2], LOW);
15	lampustate1 = false;
16	lampustate2 = false;
17	}

Selanjutnya merupakan *Source code* yang digunakan untuk mengatur penjadwalan lampu pada detik ke 33 hingga detik ke 63 yaitu mengaktifkan nilai LED hijau pada blok 1 dan juga nilai LED merah pada blok 2 serta akan mengaktifkan fungsi pembacaan LDR pada blok lampu 2 berdasarkan kondisi lampu merah 2 pada kondisi sedang aktif menyala sesuai Kode Program 5.11.

Kode Program 5.11 Fungsi Lampu mengatur detik ke 33 sampai 63

	Source Code
1	//lampu hijau 1 akan aktif selama 30detik
2	//lampu merah 2 akan aktif selama 30detik
3	else if (detiklampu >= 33 && detiklampu <63) {
4	detiklampu += 1;
5	if (detiklampu2 == 34) {
6	Serial.println("Lampu1 sedang hijau!");
7	Serial.println("Lampu2 sedang merah!");
8	}
9	Serial.println(detiklampu-33);
10	digitalWrite(lampu1[0], LOW);
11	digitalWrite(lampu1[1], LOW);
12	digitalWrite(lampu1[2], HIGH);
13	digitalWrite(lampu2[0], HIGH);
14	digitalWrite(lampu2[1], LOW);
15	digitalWrite(lampu2[2], LOW);
16	lampustate1 = false;

17	lampustate2 = true;
18	}

Selanjutnya merupakan *Source code* yang digunakan untuk mengatur penjadwalan lampu pada detik ke 33 hingga detik ke 66 yaitu mengaktifkan nilai LED kuning pada blok 1 dan juga nilai LED kuning pada blok 2 serta akan mematikan fungsi pembacaan LDR pada berdasarkan kondisi kedua lampu berada pada kondisi kuning sedang aktif menyala. Pada detik ke 66 atau satu siklus penjadwalan telah dilakukan, maka akan melakukan reset detik waktu kembali ke '0' sesuai Kode Program 5.12.

Kode Program 5.12 Fungsi Lampu mengatur detik ke 63 sampai 66

	Source Code
1	//lampu kuning 2 akan aktif selama 3detik
2	else if (detiklampu >= 63 && detiklampu <66) {
3	detiklampu += 1;
4	if (detiklampu == 64) {
5	Serial.println("Lampu1 sedang kuning!");
6	Serial.println("Lampu2 sedang kuning!");
7	}
8	Serial.println(detiklampu-63);
9	digitalWrite(lampu1[0], LOW);
10	digitalWrite(lampu1[2], LOW);
11	digitalWrite(lampu1[1], HIGH);
12	digitalWrite(lampu2[0], LOW);
13	digitalWrite(lampu2[2], LOW);
14	digitalWrite(lampu2[1], HIGH);
15	lampustate1 = false;
16	lampustate2 = false;
17	if (detiklampu >= 66) {
18	//reset penjadwalan
19	detiklampu = 0;
20	donetake1 = false;
21	donetake2 = false;
22	}
23	}
24	}

5.2.2.4 Implementasi Fungsi Servo

Source code berfungsi melakukan eksekusi task servo untuk menggerakkan atau mengatur sudut fokus kamera sebesar 25 derajat menuju blok lampu merah

yang akan aktif berdasarkan penjadwalan waktu yang ditentukan. Pertama-tama pada detik ke 31 saat sistem berjalan, servo akan diberikan nilai output analog sebesar 65 yang berupa nilai sudut untuk menggerakkan posisi fokus kamera menuju blok lampu 2 sesuai pada Kode Program 5.13.

Kode Program 5.13 Eksekusi Task Servo pada detik ke 31

	Source Code
1	void fungsiservo() {
2	detikservo += 1;
3	if(detikservo == 31){
4	myservo.write(65);
5	Serial.println("servo aktif!");
6	Serial.println("fokus kamera diganti menuju ke blok
	lampu 2");
7	}

Pada detik ke 64 saat sistem berjalan, servo akan diberikan nilai output analog sebesar 115 yang berupa nilai sudut untuk menggerakkan posisi fokus kamera menuju blok lampu 1. Pada detik ke 66 atau siklus penjadwalan telah dilakukan, maka akan melakukan reset detik waktu kembali ke '0' sesuai pada Kode Program 5.14.

Kode Program 5.14 Eksekusi Task Servo pada detik ke 64 dan 66

	Source Code
1	else if(detikservo == 64){
2	myservo.write(115);
3	Serial.println("servo aktif!");
4	Serial.println("fokus kamera diganti kembali menuju
	ke blok lampu 1");
5	}
6	else if(detikservo >= 66){
7	Serial.println("reset kembali ke awal");
8	detikservo=0;
9	}
10	}

5.2.2.5 Implementasi Fungsi Deteksi

Source code ini merupakan pembacaan pelanggaran dengan membaca nilai LDR yang aktif. Fungsi membaca nilai *output digital* pada LDR apabila nilai berubah menjadi '1' atau terjadi sebuah interrupt pelanggaran maka akan mengaktifkan buzzer dengan frekuensi 1000 Hz dengan durasi 1000 ms atau 1

detik. Pada Kode Program 5.15 merupakan fungsi pendeteksi pada blok lampu 1 dimana terdapat kondisi pembacaan yaitu kondisi lampu merah 1 berada pada kondisi merah, pengecekan perulangan pembacaan deteksi untuk menghindari *spam*, serta pembacaan nilai digital pada sensor LDR 1 harus bernilai '1' atau terdeteksi adanya gangguan.

Kode Program Program 5.15 Fungsi Deteksi pada blok lampu 1

	Source Code
1	void deteksi() {
2	if (!donetake1 && lampustate1 && LDR1 == 1) {
3	tone(buzzer, 1000, 1000);
4	takeCamera();
5	detikservo += 5;
6	detiklampu += 5;
7	detiklampu2 += 5;
8	previousMillisLampu += 5000;
9	previousMillisServo += 5000;
10	donetake1 = true;
11	}

Pada Kode Program 5.16 merupakan fungsi pendeteksi pada blok lampu 2 dimana terdapat kondisi pembacaan yaitu kondisi lampu merah 2 berada pada kondisi merah, pengecekan perulangan pembacaan deteksi untuk menghindari *spam*, serta pembacaan nilai digital pada sensor LDR 2 harus bernilai '1' atau terdeteksi adanya gangguan.

Kode Program 5.16 Fungsi Deteksi pada blok lampu 2

	Source Code
1	else if (!donetake2 && lampustate2 && LDR2 == 1) {
2	tone(buzzer, 1000, 1000);
3	takeCamera();
4	detiklampu += 5;
5	detikservo += 5;
6	detiklampu2 += 5;
7	previousMillisLampu += 5000;
8	previousMillisServo += 5000;
9	donetake2 = true;
10	}
11	}

5.2.2.6 Implementasi Fungsi Kamera

Source code dibawah merupakan fungsi untuk pengaktifkan kamera. Pertama akan melakukan inisialisasi ukuran pengambilan gambar yang akan disimpan, lalu mengaktifkan nilai digital pengambilan gambar atau *snapshot* pada kamera. Kemudian dilakukan penamaan dari hasil gambar yang telah diambil, penamaan data gambar dilakukan dengan format penulisan "IMAGE[angka].JPG" disimpan dengan format .jpg. apabila kartu SD card tidak dapat terdeteksi maka operasi akan diberhentikan sesuai pada Kode Program 5.17 yaitu Fungsi Kamera.

Kode Program 5.17 Fungsi Kamera

	Source Code
1	void takeCamera() {
2	cam.setImageSize(VC0706_320x240);
3	if (cam.takePicture()) {
4	digitalWrite(lampucamera, HIGH);
5	} else {
6	}
7	char filename[13];
8	strcpy(filename, "IMAGE00.JPG");
9	for (int i = 0; i < 100; i++) {
10	filename[5] = '0' + i / 10;
11	filename[6] = '0' + i % 10;
12	if (!SD.exists(filename)) {
13	break;
14	}
15	}

Berikutnya merupakan fungsi untuk penulisan atau penyimpanan data gambar yang telah diambil. Pertama fungsi akan membuka micro sd card lalu melakukan pengukuran panjang dari frame gambar dari file gambar yang telah diambil. Terdapat buffer yang digunakan untuk membaca tiap byte pada file gambar yang berasal dari kamera. Setelah inisialisasi dilakukan maka perulangan penulisan tiap byte file dengan menggunakan fungsi `bytesToRead` dengan setting maksimal pembacaan 64bytes, perulangan dilakukan sampai akhir dari panjang frame gambar. Kemudian akan melakukan penutupan file gambar yang telah disimpan dan menjalankan pembacaan video kembali pada kamera sesuai dengan Kode 5.18.

Kode Program 5.18 Penyimpanan File Gambar

	Source Code
1	File imgFile = SD.open(filename, FILE_WRITE);
2	uint16_t jpglen = cam.frameLength();

```
3      while (jpglen > 0) {  
4          uint8_t *buffer;  
5          uint8_t bytesToRead = min(64, jpglen);  
6          buffer = cam.readPicture(bytesToRead);  
7          imgFile.write(buffer, bytesToRead);  
8          jpglen -= bytesToRead;  
9      }  
10     imgFile.close();  
11     cam.resumeVideo();  
12     digitalWrite(lampucamera, LOW);  
13 }
```



BAB 6 PENGUJIAN DAN ANALISIS

Pada bab pengujian dan analisis akan membahas tentang pengujian dari sistem yang telah dibuat disertai analisis. Terdapat tiga pengujian yang dilakukan yaitu pengujian fungsionalitas penjadwalan dari sistem, pengujian respon rata-rata waktu sistem terhadap *trigger* pelanggaran yang terjadi dan pengujian real time pada sistem dengan membandingkan ketepatan waktu yang berjalan pada sistem dengan pengukuran waktu yang nyata.

6.1 Pengujian *Scheduling* Sistem

6.1.1 Tujuan Pengujian

Pengujian *Scheduling* sistem dilakukan untuk menguji kemampuan fungsionalitas pada sistem untuk menjalankan *task* sesuai dengan waktu yang ditentukan agar sistem mampu bekerja secara *real time*. Pengujian *scheduling* juga akan melakukan perbandingan waktu eksekusi sistem dibandingkan dengan pengukuran waktu yang nyata. Terdapat tiga buah penjadwalan yang dilakukan yaitu penjadwalan lampu lalu lintas 1 dan lampu lalu lintas 2 yang terdiri dari lampu merah, kuning dan hijau serta penjadwalan pada servo untuk mengatur fokus lampu bergantian menuju blok lampu dengan status merah yang akan aktif.

6.1.2 Pelaksanaan Pengujian

Pengujian dilakukan dengan menghubungkan perangkat keras sistem yaitu mikrokontroler arduino Mega ke sumber daya atau laptop. Pada perangkat lunak Arduino IDE digunakan untuk melakukan *serial monitoring* pada sistem yang berjalan peneliti mengamati waktu eksekusi sebuah task dan membandingkan dengan waktu di dunia nyata. Task yang berjalan harus mampu bekerja sesuai jadwal yang ditentukan.

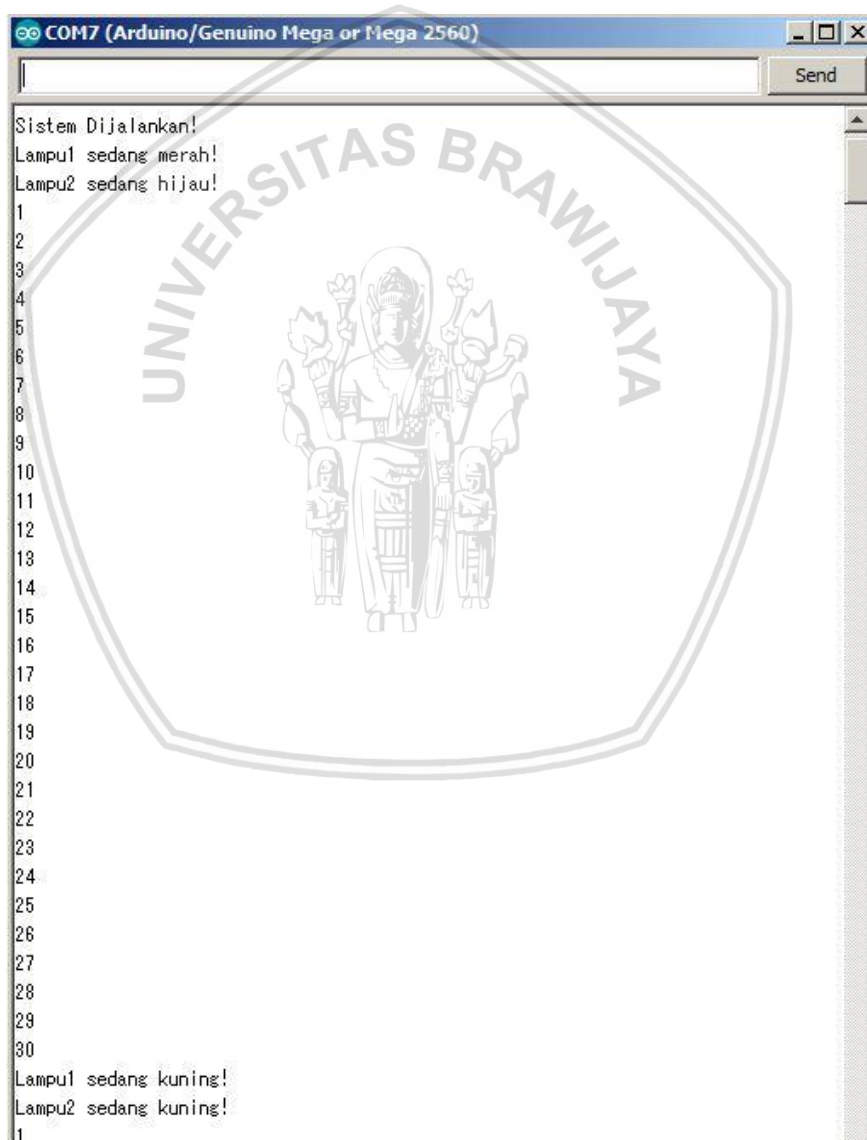
6.1.3 Prosedur Pengujian

Pengujian akan dilakukan dengan langkah-langkah sesuai dengan prosedur berikut ini.

1. Menghubungkan Mikrokontroler dengan laptop untuk mendapat sumber daya.
2. Menjalankan Aplikasi Arduino IDE serta mengupload program ke dalam mikrokontroler Arduino Mega.
3. Menjalankan Serial Monitor.
4. Mengamati waktu berjalannya eksekusi *task-task* yaitu lampu dan servo harus mampu berjalan sesuai jadwal yang ditentukan.
5. Melakukan analisis perbandingan dimulainya waktu eksekusi tiap task hingga berakhir dengan pengukuran waktu yang nyata.

6.1.4 Hasil Pengujian

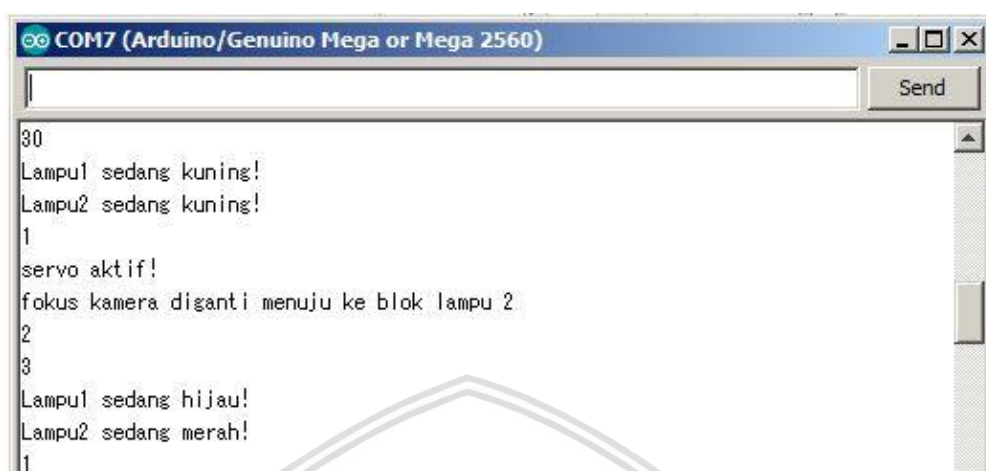
Pengujian dilakukan dengan menjalankan serial Monitor yang terdapat pada aplikasi Arduino IDE. Selanjutnya melakukan pengujian penjadwalan nyala lampu LED sebagai indikasi lampu lalu lintas dan juga servo sebagai penggerak fokus kamera mampu aktif sesuai dengan penjadwalan yang dilakukan. Pertama lampu merah 1 dan lampu hijau 2 akan aktif pada 30 detik pertama lalu kedua lampu kuning akan aktif selama 3 detik, lalu dilanjutkan dengan lampu hijau 1 dan lampu merah 2 akan aktif selama 30 detik dan dilanjutkan dengan kedua lampu kuning akan aktif selama 3 detik kemudian perulangan akan kembali ke awal lampu merah 1 dan lampu hijau akan aktif. Servo akan aktif pada detik 2 pada saat lampu kuning sedang aktif, servo akan menggerakkan fokus dari modul kamera menuju ke blok lampu dengan status lampu merah yang akan aktif.



Gambar 6.1 Hasil pengujian pada 30 detik pertama

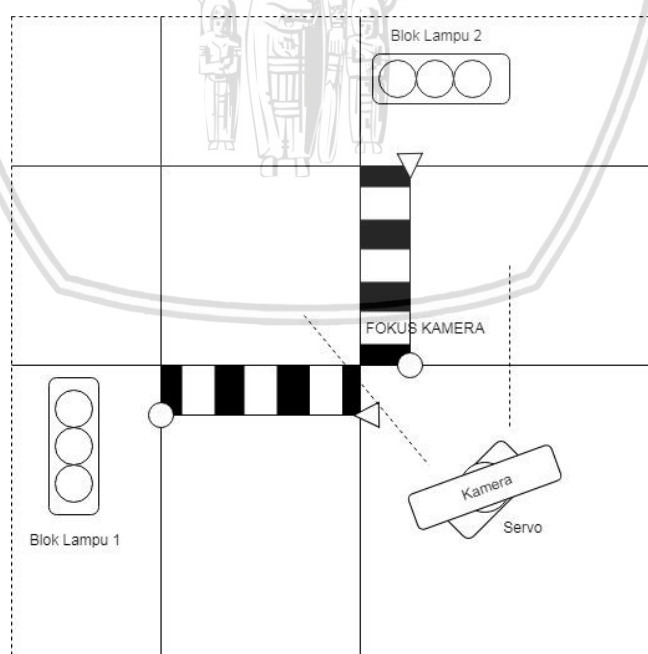
Pada Gambar 6.1 dilakukan pengujian mengamati 30 detik pertama pada saat program dijalankan. LED merah pada blok lampu 1 akan aktif menyala dan LED

hijau pada blok lampu 2 akan aktif menyala pada detik ke 1 sampai detik ke 30 dan mengaktifkan pembacaan pelanggaran pada blok lampu 1. Setelah 30 detik berlalu selanjutnya kedua lampu kuning akan aktif selama 3 detik.



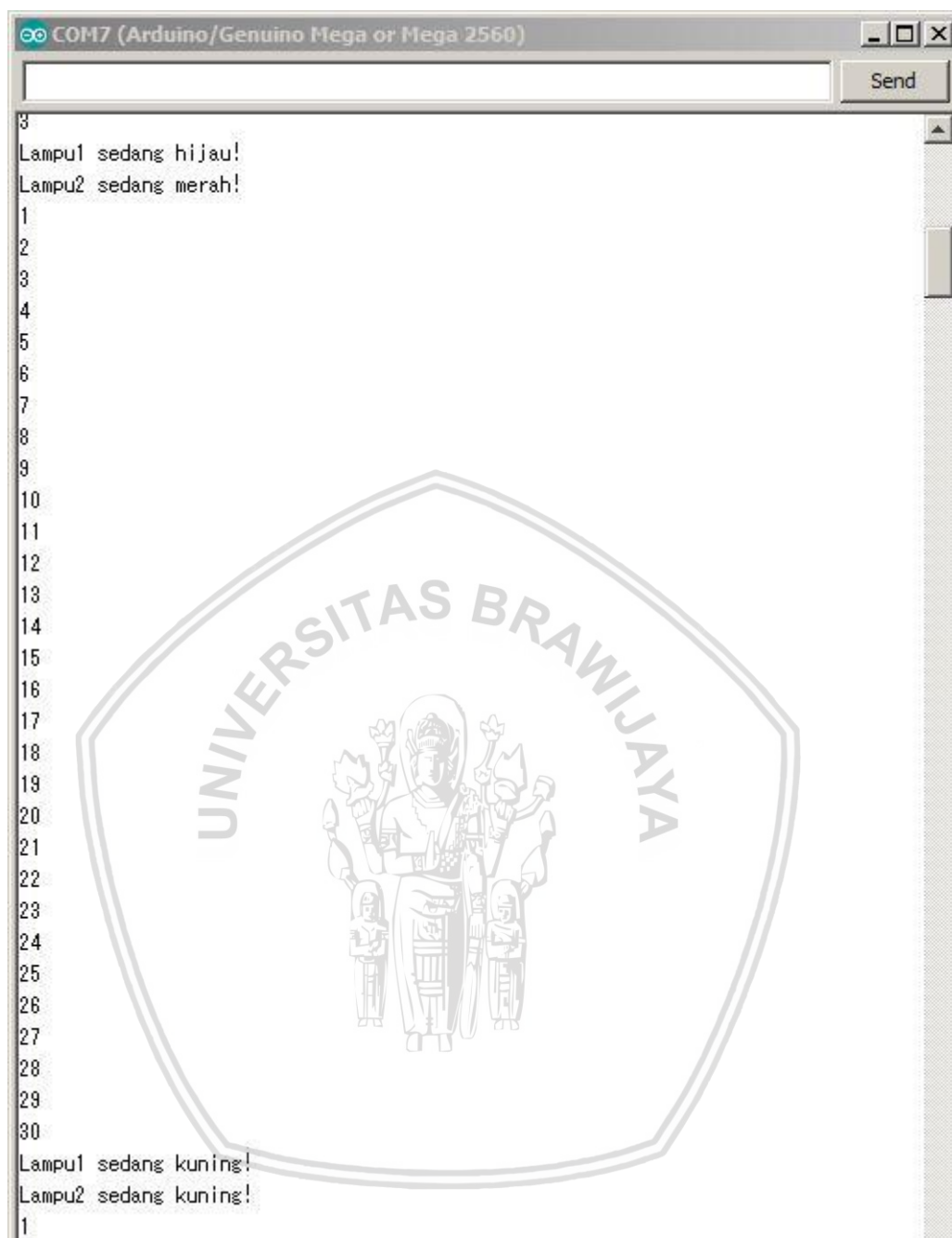
Gambar 6.2 Hasil pengujian pada detik ke 30 sampai detik ke 33

Pada Gambar 6.2 setelah 30 detik pertama berlalu. Selanjutnya kedua lampu kuning pada kedua blok lampu akan aktif selama 3 detik. Pembacaan pelanggaran pada blok 1 dan blok 2 akan dinon-aktifkan. Pada detik ke 1 saat lampu kuning sedang berjalan, servo akan menggerakkan fokus dari kamera menuju ke blok lampu 2 atau servo berada pada sudut 115 derajat (25 derajat ke kanan) sebelum status lampu merah pada blok lampu 2 akan Diaktifkan seperti yang terlihat pada Gambar 6.3.



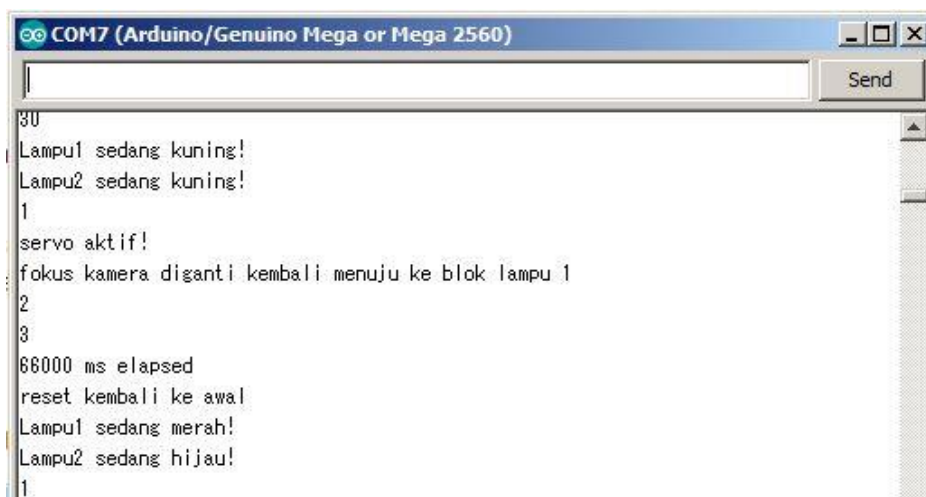
Gambar 6.3 Posisi Fokus Kamera pada detik ke 31

Setelah detik ke 33 berlalu, pada detik 34 lampu 1 akan mulai aktif berstatus hijau dan lampu 2 berstatus merah.



Gambar 6.4 Hasil pengujian pada detik ke 33 sampai detik ke 63

Pada Gambar 6.4 mengamati detik ke 34 yang berlalu setelah program dijalankan. LED hijau pada blok lampu 1 akan aktif menyala dan LED merah pada blok lampu 2 akan aktif menyala selama 30 detik serta mengaktifkan pembacaan pelanggaran pada blok lampu 2. Setelah 30 detik berlalu selanjutnya yaitu pada detik ke 64 kedua lampu kuning akan diaktif selama 3 detik.



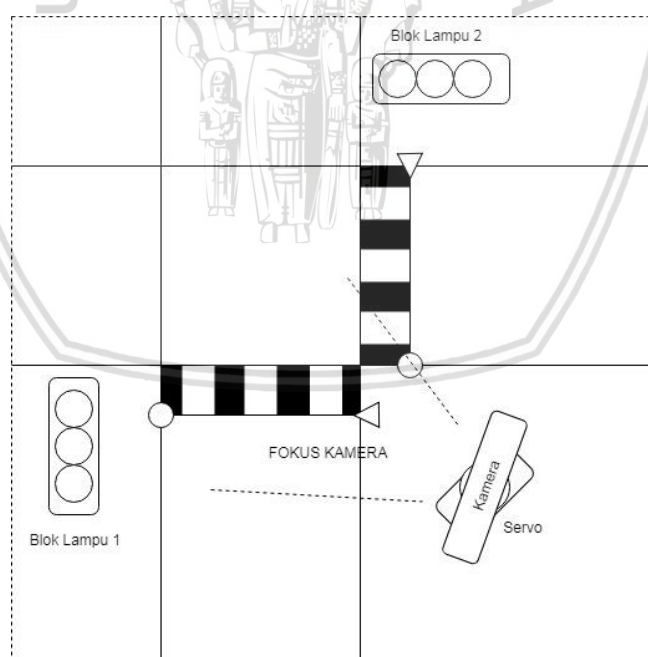
```

COM7 (Arduino/Genuino Mega or Mega 2560)
30
Lampu1 sedang kuning!
Lampu2 sedang kuning!
1
servo aktif!
fokus kamera diganti kembali menuju ke blok lampu 1
2
3
66000 ms elapsed
reset kembali ke awal
Lampu1 sedang merah!
Lampu2 sedang hijau!
1

```

Gambar 6.5 Hasil pengujian pada detik ke 63 sampai detik ke 66

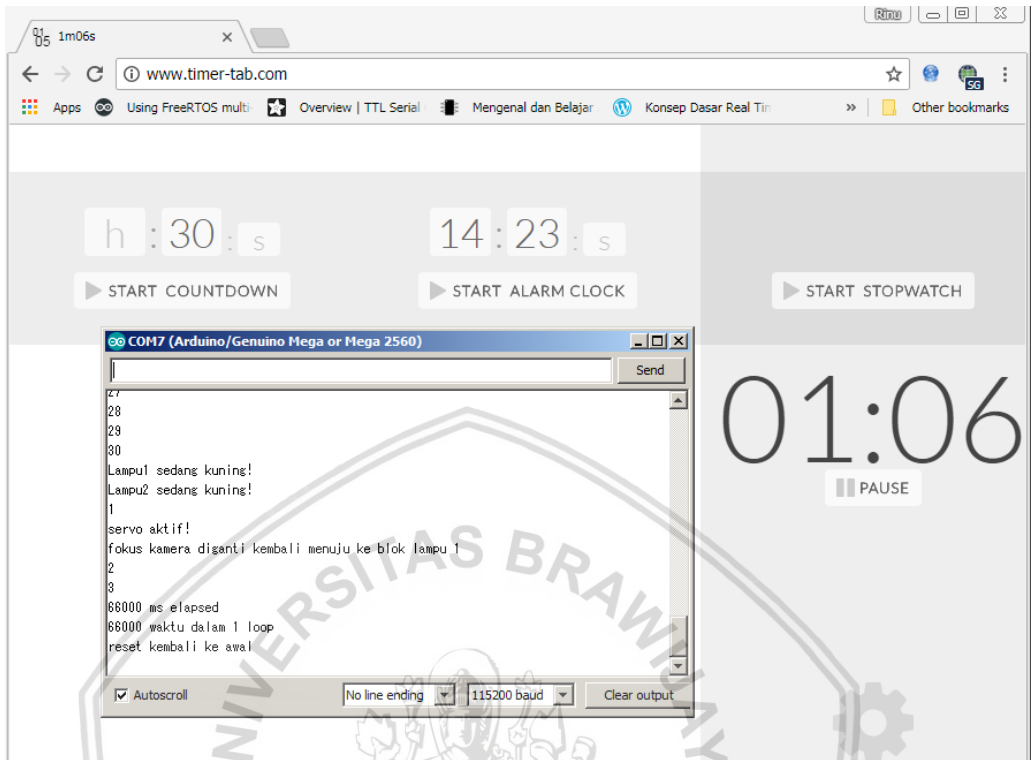
Pada Gambar 6.5 setelah 63 detik pertama berlalu atau lampu hijau pada blok lampu 1 dan lampu merah pada blok lampu 2 telah aktif menyala selama 30 detik. Selanjutnya pada detik kedua lampu kuning pada kedua blok lampu akan aktif selama 3 detik. Pembacaan pelanggar pada blok 1 dan blok 2 akan dinon-aktifkan. Pada detik ke 64 saat lampu kuning sedang berjalan, servo akan menggerakkan fokus dari kamera kembali menuju ke blok lampu 1 atau servo berada pada sudut 65 derajat (25 derajat ke kiri) sebelum status lampu merah pada blok lampu 1 akan Diaktifkan seperti yang terlihat pada Gambar 6.6.



Gambar 6.6 Posisi Fokus Kamera pada detik ke 64

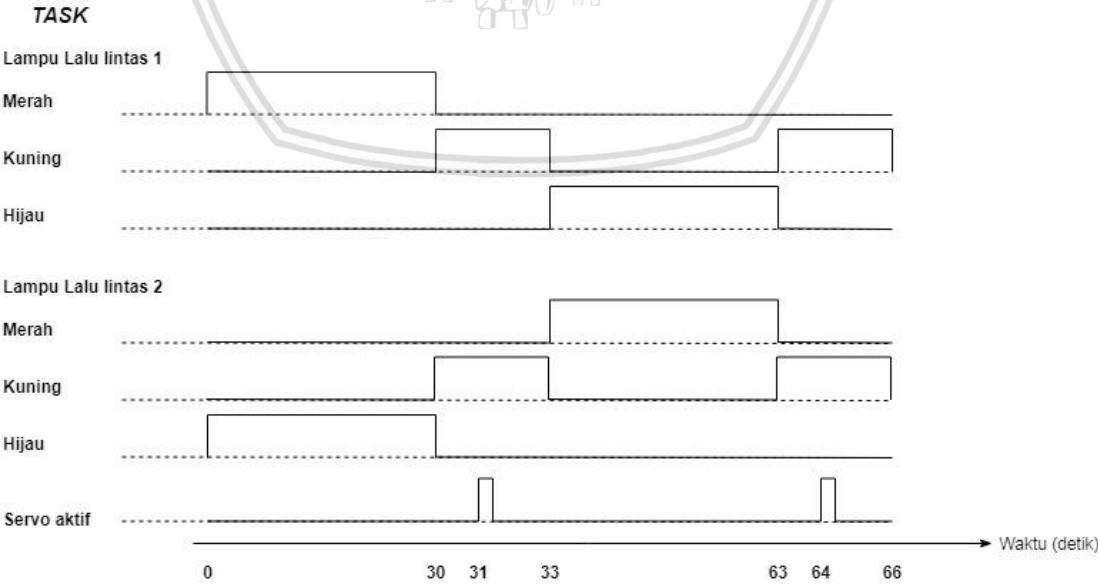
Pada detik 66 perulangan akan dilakukan kembali pada detik ke 0. Pada *serial monitor* terdapat hasil uji pengukuran lama waktu satu siklus proses yang telah dijalankan dengan hasil 66000 ms atau 66 detik. Kemudian dilakukan

perbandingan dengan hasil pengukuran waktu secara nyata dengan menggunakan online timer. Sesuai pada Gambar 6.7.



Gambar 6.7 Perbandingan waktu eksekusi sistem dengan waktu nyata

Pada hasil pengujian terdapat kesamaan waktu yang dibutuhkan dalam melakukan pekerjaan dalam satu siklus yaitu 66 detik. Pada gambar 6.8 merupakan diagram waktu hasil pengukuran pada pengujian *scheduling* yang telah dilakukan.



Gambar 6.8 Diagram Waktu Pengujian *Scheduling*

6.1.5 Analisis Hasil Pengujian

Setelah pengujian *Scheduling* telah dilakukan, dapat disimpulkan didapatkan seperti yang terlihat pada Tabel 6.1. Keberhasilan pengujian diberi tanda *checklist* (✓) sedangkan pengujian yang tidak berhasil diberi tanda *cross* (x). Berdasarkan hasil tersebut terlihat bahwa sistem berhasil menjalankan fungsi sesuai dengan tujuannya dan melakukan penjadwalan sesuai waktu yang nyata.

Tabel 6.1 Hasil Pengujian Penjadwalan pada Sistem

Nomor Pengujian	Nama Pengujian	Perhitungan detik (s) ke pada		Fungsi Berjalan
		Sistem	Waktu Nyata	
1	Lampu Merah 1 aktif Lampu Hijau 2 aktif	0	0	✓
2	Lampu Kuning 1 aktif Lampu Kuning 2 aktif	30	30	✓
3	Fokus kamera bergerak ke lampu 2	31	31	✓
4	Lampu Hijau 1 aktif Lampu Merah 2 aktif	33	33	✓
5	Lampu Kuning 1 aktif Lampu Kuning 2 aktif	63	63	✓
6	Fokus kamera bergerak ke lampu 1	64	64	✓

Berdasarkan Pengujian pada Tabel 6.1 sistem mampu menjalankan seluruh tugas fungsionalitas pada fungsi penjadwalan tepat dengan deadline waktu pengukuran yang nyata.

6.2 Pengujian Deteksi Pelanggar

6.2.1 Tujuan Pengujian

Pengujian Deteksi Pelanggar digunakan untuk menguji fungsionalitas pada sistem dalam melakukan pendeteksi pelanggaran dengan menggunakan konsep *laser security system*. Saat melakukan pengujian diharapkan sistem dapat memberikan output respon *alarm* yang tepat yaitu buzzer dan kamera aktif serta data gambar dapat disimpan ke dalam micro sd card yang terpasang pada saat kondisi blok lampu berstatus merah sedang aktif. serta melakukan perhitungan hasil eksekusi waktu rata-rata yang diperlukan oleh sistem untuk mengaktifkan alarm pelanggaran.

6.2.2 Pelaksanaan Pengujian

Pengujian dilakukan dengan menghubungkan perangkat keras sistem yaitu mikrokontroler arduino Mega ke sumber daya atau laptop. Pada perangkat lunak Arduino IDE digunakan untuk melakukan *serial monitoring* pada sistem yang berjalan peneliti dapat melihat fungsi-fungsi yang berjalan pada sistem. Fungsi yang berjalan harus mampu berfungsi sesuai dengan tujuan dari kebutuhan fungsionalnya.

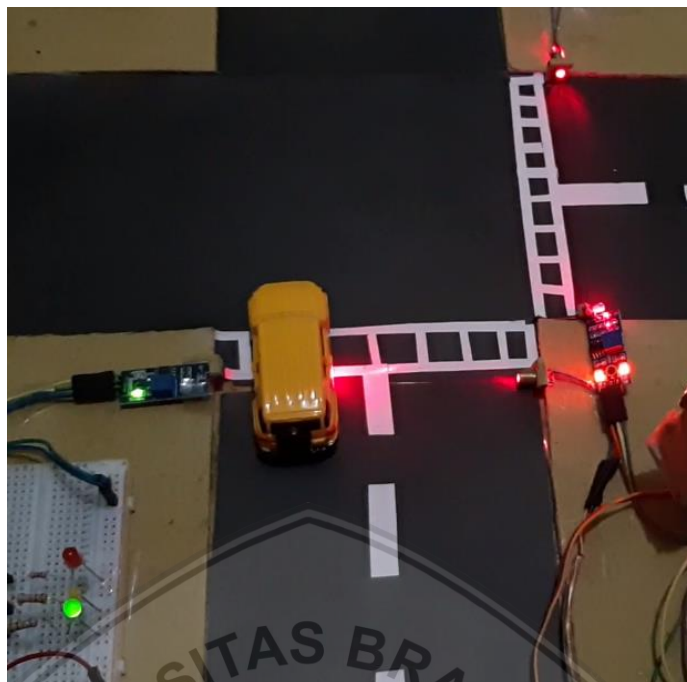
6.2.3 Prosedur Pengujian

Pengujian akan dilakukan dengan langkah-langkah sesuai dengan prosedur berikut ini.

1. Memasangkan micro SD Card ke dalam Modul SD Card yang terhubung pada Arduino.
2. Menyambungkan Arduino Mega dengan laptop agar menyala.
3. Menjalankan Aplikasi Arduino IDE serta mengupload program ke dalam mikrokontroler Arduino Mega.
4. Menjalankan Serial Monitor.
5. Melakukan pengamatan dengan melakukan *trigger* pelanggaran pada saat blok lampu dengan status lampu kuning atau hijau sedang aktif.
6. Melakukan pengamatan dengan melakukan *trigger* pelanggaran pada saat blok lampu merah berstatus aktif untuk mengaktifkan alarm.
7. Mengulang mengujian trigger pelanggar beberapa kali dengan menggunakan 5 objek serta dengan 3 resolusi ukuran gambar yang berbeda.
8. Melihat output hasil *snapshot* dari kamera VC0706 pada micro sd card dengan menggunakan sd card reader untuk mengetahui hasil gambar pada saat terjadinya pelanggaran.
9. Melakukan analisis respon waktu dari *alarm* pelanggaran yang telah dihasilkan oleh sistem pada saat terjadinya pelanggar.

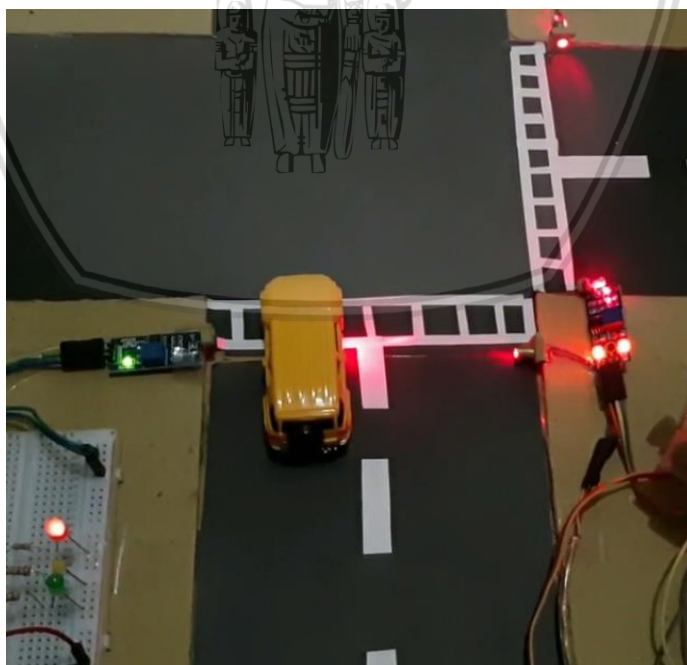
6.2.4 Hasil Pengujian

Pengujian dilakukan dengan menjalankan serial Monitor yang terdapat pada aplikasi Arduino IDE. Selanjutnya melakukan pengujian Melakukan *trigger* pelanggaran atau mengganggu pembacaan *laser diode* pada sensor LDR pada saat blok dengan status lampu kuning atau hijau sedang aktif, sistem tidak memberikan respon *alarm* pada saat terjadinya gangguan pembacaan LDR. Karena sistem hanya akan mampu membaca deteksi pelanggaran apabila blok lampu berstatus lampu merah sedang aktif seperti yang terdapat pada Gambar 6.9.



Gambar 6.9 Kondisi Sensor Deteksi tidak aktif

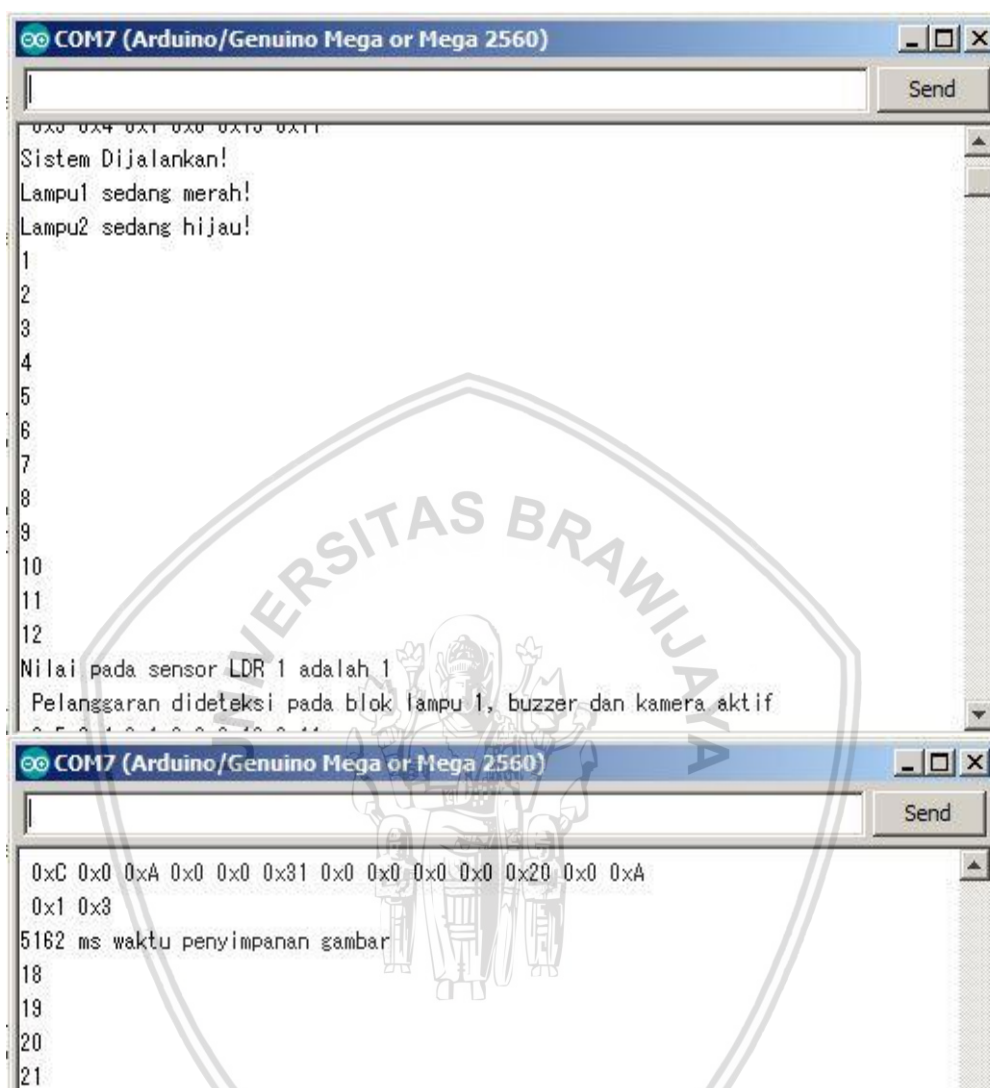
Pada pengujian selanjutnya akan melakukan pengujian pelanggaran pada suatu blok lampu dengan status lampu merah sedang aktif untuk mengamati respon sistem yang menghasilkan output alarm. Kondisi sistem akan melakukan pendeteksian pelanggar seperti yang digambarkan seperti yang ada pada Gambar 6.10.



Gambar 6.10 Kondisi Sensor Deteksi aktif

Pengujian yang pertama dilakukan pengamatan eksekusi pada serial monitor pada Arduino IDE dengan menggunakan setting ukuran pengambilan gambar

kamera sebesar 320x240 pixel, sistem mampu melakukan eksekusi alarm tepat sesuai pada saat kondisi pelanggaran terdeteksi yaitu pada saat lampu merah sedang aktif dan nilai sensor LDRnya bernilai '1' seperti pada Gambar 6.11



Gambar 6.11 Hasil Pengujian alarm pendeteksi

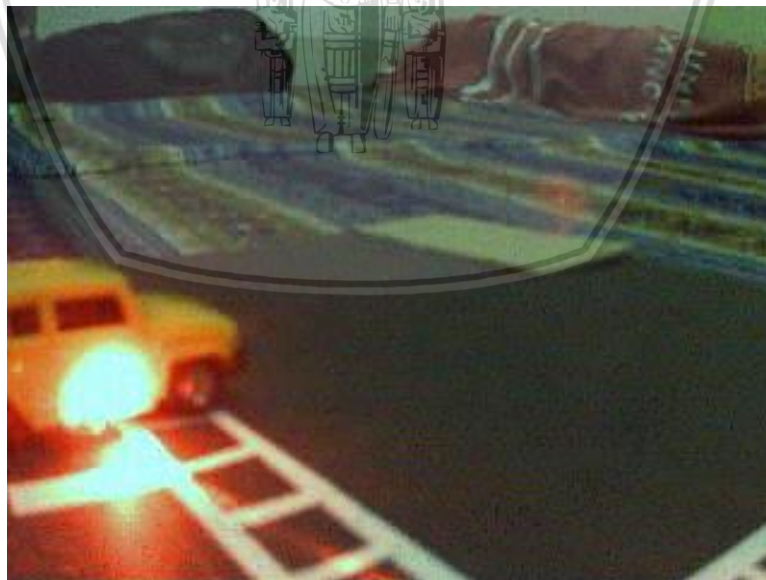
Lalu Pengujian deteksi akan dilakukan dengan menggunakan 5 objek yang berbeda serta menggunakan 3 ukuran gambar pengambilan yang berbeda. Sistem yang diharapkan akan mampu memberikan respon alarm dengan mengaktifkan buzzer dan fungsi kamera. Setelah dilakukan pengujian sistem mampu menghasilkan respon alarm sebanyak 5 kali sesuai dengan 5 objek berbeda yang diujikan dengan menggunakan resolusi gambar 640x480 pixel, 320x240 pixel serta 160x120 pixel. Kemudian dilakukan pengecekan hasil gambar yang tersimpan di dalam micro SD card yang tertanam pada modul SD card dengan menggunakan *SD Card Reader* yang dihubungkan dengan laptop peneliti. Terdapat 15 hasil gambar pelanggaran dengan menggunakan 5 objek yang berbeda dan 3 ukuran resolusi yang berbeda. Gambar dibawah merupakan hasil pengujian yang telah dilakukan.

1. Deteksi Pelanggar dengan menggunakan Objek 1

Pada pengujian dengan menggunakan objek yang pertama yaitu menggunakan miniatur mobil pada blok lampu 1 dengan menggunakan resolusi 640x480 pixel seperti pada Gambar 6.12. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 19560 ms.



Gambar 6.12 Hasil pengujian pedeteksi pelanggar dengan objek 1 menggunakan resolusi 640x480



Gambar 6.13 Hasil pengujian pedeteksi pelanggar dengan objek 1 menggunakan resolusi 320x240

lalu pada pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 320x240 pixel seperti pada Gambar 6.13. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar

sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 5122 ms.



Gambar 6.14 Hasil pengujian pedeteksi pelanggar dengan objek 1 menggunakan resolusi 160x120

lalu pengujian deteksi dengan menggunakan objek yang sama selanjutnya dengan menggunakan resolusi 160x120 pixel seperti pada Gambar 6.14. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 1579 ms.

2. Deteksi Pelanggar dengan menggunakan Objek 2



Gambar 6.15 Hasil pengujian pedeteksi pelanggar dengan objek 2 menggunakan resolusi 640x480

Pada pengujian dengan menggunakan objek yang kedua yaitu menggunakan buku pada blok lampu 2 dengan resolusi 640x480 pixel seperti pada Gambar 6.15. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan

mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 19701 ms.



Gambar 6.16 Hasil pengujian pedeteksi pelanggar dengan objek 2 menggunakan resolusi 320x240

lalu pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 320x240 pixel seperti pada Gambar 6.16. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 5170 ms.



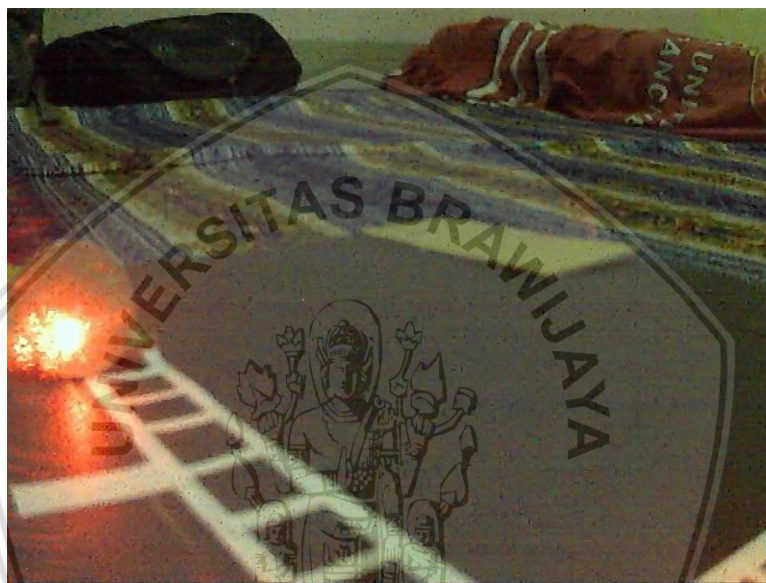
Gambar 6.17 Hasil pengujian pedeteksi pelanggar dengan objek 2 menggunakan resolusi 160x120

lalu pengujian deteksi dengan menggunakan objek yang sama selanjutnya dengan menggunakan resolusi 160x120 pixel seperti pada Gambar 6.17. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil

gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 1573 ms.

3. Deteksi Pelanggar dengan menggunakan Objek 3

Pada pengujian dengan menggunakan objek yang ketiga yaitu menggunakan koin pada blok lampu 1 dengan menggunakan resolusi gambar 640x480 pixel seperti pada Gambar 6.18. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 19056 ms atau 19,05 detik.



Gambar 6.18 Hasil pengujian pedeteksi pelanggar dengan objek 3 menggunakan resolusi 640x480

lalu pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 320x240 pixel seperti pada Gambar 6.19. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 5132 ms.



Gambar 6.19 Hasil pengujian pedeteksi pelanggar dengan objek 3 menggunakan resolusi 320x240

kemudian pengujian deteksi dengan menggunakan objek yang sama selanjutnya dengan menggunakan resolusi 160x120 pixel seperti pada Gambar 6.20. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 1606 ms.



Gambar 6.20 Hasil pengujian pedeteksi pelanggar dengan objek 3 menggunakan resolusi 160x120

4. Deteksi Pelanggar dengan menggunakan Objek 4

Pada pengujian dengan menggunakan objek yang keempat yaitu menggunakan smartphone pada blok lampu 2 dengan resolusi 640x480 seperti pada Gambar 6.21. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi untuk mengaktifkan alarm adalah 19587 ms



Gambar 6.21 Hasil pengujian pedeteksi pelanggar dengan objek 4 menggunakan resolusi 640x480

lalu pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 320x240 pixel seperti pada Gambar 6.22. Sistem mampu memberikan respon *alarm*. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 5116 ms.



Gambar 6.22 Hasil pengujian pedeteksi pelanggar dengan objek 4 dengan resolusi 320x240

kemudian pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 160x120 pixel seperti pada Gambar 6.23. Sistem mampu memberikan respon *alarm*. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 1580 ms.



Gambar 6.23 Hasil pengujian pedeteksi pelanggar dengan objek 4 dengan resolusi 160x120

5. Deteksi Pelanggar dengan menggunakan Objek 5

Pada pengujian dengan menggunakan objek yang kelima yaitu menggunakan tangan pada blok lampu 1 dengan resolusi 640x480 seperti pada Gambar 6.24. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 19721 ms.



Gambar 6.24 Hasil pengujian pedeteksi pelanggar dengan objek 5 dengan resolusi 640x480

lalu pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 320x240 pixel seperti pada Gambar 6.25. Sistem mampu memberikan respon *alarm*. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 5237 ms.



Gambar 6.25 Hasil pengujian pedeteksi pelanggar dengan objek 5 dengan resolusi 320x240

kemudian pengujian deteksi dengan menggunakan objek yang sama dengan menggunakan resolusi 160x120 pixel seperti pada Gambar 6.26. Sistem mampu memberikan respon *alarm* dengan mengaktifkan buzzer dan mengambil gambar sekaligus menyimpan ke dalam micro sd card. Waktu eksekusi yang diperlukan untuk mengaktifkan alarm adalah 1560 ms.



Gambar 6.26 Hasil pengujian pedeteksi pelanggar dengan objek 5 menggunakan resolusi 160x120

6.2.5 Analisis Hasil Pengujian

Setelah pengujian deteksi telah dilakukan, didapatkan hasil pengujian fungsionalitas seperti yang terlihat pada Tabel 6.2. Keberhasilan pengujian diberi tanda *checklist* (✓) sedangkan pengujian yang tidak berhasil diberi tanda *cross* (x). Berdasarkan hasil tersebut terlihat bahwa sistem mampu menjalankan seluruh fungsi alarm sesuai dengan tujuan dari sistem.

Tabel 6.2 Hasil Pengujian Fungsionalitas Deteksi Pelanggaran

Nomor Pengujian	Objek	Pengujian Fungsional		
		Buzzer Aktif	Snapshot	Menyimpan Gambar
1	Miniatur mobil	✓	✓	✓
2	Buku	✓	✓	✓
3	Koin	✓	✓	✓
4	Smartphone	✓	✓	✓
5	Tangan	✓	✓	✓

Berikutnya pada Tabel 6.3 merupakan pengujian perhitungan waktu eksekusi alarm yang telah dilakukan dengan menggunakan lima objek yang berbeda serta pada tiga resolusi yang berbeda-beda.

Tabel 6.3 Data Rata-rata Waktu Eksekusi Alarm Pelanggar

Nomor Pengujian	Objek	Waktu Eksekusi Alarm (ms) dengan menggunakan resolusi		
		640x480	320x240	160x120
1	Miniatur mobil	19560	5122	1579
2	Buku	19701	5170	1573
3	Koin	19056	5132	1606
4	Smartphone	19587	5116	1580
5	Tangan	19721	5237	1562
RATA-RATA		19525	5155,4	1580

Berdasarkan hasil pengujian dengan 5 objek yang berbeda yang terdapat pada tabel 6.3 diatas, rata-rata waktu eksekusi yang dibutuhkan sistem untuk mengaktifkan alarm yang terdiri jadi menyalakan buzzer, mengambil gambar serta menuliskan data gambar dengan resolusi 640x480 pada micro sd card adalah 19525 ms. Sedangkan menggunakan resolusi gambar 320x240 waktu rata-rata eksekusi yang diperlukan adalah 5155,4 ms dan menggunakan resolusi gambar 160x120 waktu rata-rata eksekusi yang diperlukan adalah 1580 ms.

6.3 Pengujian Ketepatan Waktu Eksekusi dalam satu siklus penjadwalan

6.3.1 Tujuan Pengujian

Pengujian Ketepatan waktu eksekusi yang dihasilkan oleh sistem *real time* yang menggunakan Millis pada sistem yang telah dirancang oleh peneliti. Untuk mengetahui seberapa tepat dan akurat hasil waktu eksekusi yang telah berjalan pada saat satu siklus task yang telah dieksekusi terhadap deadline yang ditentukan yaitu 66 detik perhitungan waktu yang nyata.

6.3.2 Pelaksanaan Pengujian

Sebelum pengujian dilakukan menghubungkan perangkat keras sistem yaitu mikrokontroler arduino Mega ke sumber daya atau laptop. Pada perangkat lunak Arduino IDE akan diuji dengan *source code* yang sudah diberikan fungsi *timer* berupa *millis()* dan akan diupload menuju mikrokontroler dan akan melakukan pengamatan *serial monitoring* pada *source code* yang diuji.

Pengujian dilakukan dengan membandingkan tiap waktu eksekusi proses yang dibutuhkan pada tiap satu loop program atau dalam kurun waktu 66 detik. Yang terbagi atas 30 detik lampu merah dan hijau, 3 detik lampu kuning, 30 detik lampu hijau dan merah dan 3 detik lampu kuning. Pengujian dilakukan selama 10 kali *loop* program berjalan.

6.3.3 Prosedur Pengujian

Pengujian akan dilakukan dengan langkah-langkah sesuai dengan prosedur berikut ini.

1. Menghubungkan Mikrokontroler ke laptop.
2. Membuka aplikasi Arduino IDE.
3. Melakukan upload program sistem peneliti yang menggunakan fungsi *millis()*
4. Menjalankan serial monitor.
5. Melakukan pengamatan pada sistem dan serial monitoring selama 10 loop sistem berjalan
6. Melakukan analisis berdasarkan hasil ketepatan waktu pada penjadwalan task.

6.3.4 Hasil Pengujian

Pengujian pertama dilakukan pada program sistem peneliti yang memanfaatkan fungsi *millis()* dengan menjalankan serial Monitor yang terdapat pada aplikasi Arduino IDE. Selanjutnya melakukan pengamatan pada *serial monitor* pada saat sistem dijalankan. Pengamatan akan dilakukan selama satu siklus loop program dibandingkan dengan 66 detik perhitungan nyata.



```

COM7 (Arduino/Genuino Mega or Mega 2560)
27
28
29
30
Lampu1 sedang kuning!
Lampu2 sedang kuning!
1
servo aktif!
fokus kamera diganti kembali menuju ke blok lampu 1
2
3
loop ke 1
66000 waktu dalam satu loop
waktu yang berjalan :66000 ms elapsed
reset kembali ke awal

```

Gambar 6.27 Hasil waktu eksekusi satu siklus program pada loop pertama

Pada pengamatan pada serial monitor saat menjalankan sistem, pada loop ke satu sistem mampu melakukan waktu eksekusi selama 66000 ms atau 66 detik seperti pada Gambar 6.27. Sistem mampu memberikan respon waktu dalam menjalankan task dalam satu siklus secara akurat tanpa adanya *delay* sesuai dengan *deadline* waktu yang diberikan.



```

COM7 (Arduino/Genuino Mega or Mega 2560)
0.11 ms waktu penyimpanan gambar
28
29
30
Lampu1 sedang kuning!
Lampu2 sedang kuning!
1
servo aktif!
fokus kamera diganti kembali menuju ke blok lampu 1
2
3
loop ke 5
66046 waktu dalam satu loop
waktu yang berjalan :330046 ms elapsed
reset kembali ke awal

```

Gambar 6.28 Hasil waktu eksekusi satu siklus program pada loop kelima

Pada pengamatan selanjutnya pada loop kelima atau 330 detik telah berjalan, sistem mampu melakukan waktu eksekusi satu loop selama 66046 ms atau 66,046 detik seperti yang ada pada Gambar 6.28. Pada loop kelima sistem

memberikan respon waktu 0,046 detik lebih lambat berdasarkan *deadline* waktu task dalam satu siklus yaitu 66 detik.



```

COM7 (Arduino/Genuino Mega or Mega 2560)
27
28
29
30
Lampu1 sedang kuning!
Lampu2 sedang kuning!
1
servo aktif!
fokus kamera diganti kembali menuju ke blok lampu 1
2
3
loop ke 10
66000 waktu dalam satu loop
waktu yang berjalan :660046 ms elapsed
reset kembali ke awal

```

Gambar 6.29 Hasil waktu eksekusi satu siklus program pada loop kesepuluh

Pada pengujian yang terakhir yaitu pada loop kesepuluh atau 660 detik telah berjalan, sistem mampu melakukan waktu eksekusi satu loop selama 66000 ms atau 66 detik seperti yang ada pada Gambar 6.29. Sistem kembali mampu memberikan respon waktu penyelesaian menjalankan task dalam satu siklus secara akurat sesuai dengan *deadline* waktu yang diberikan.

6.3.5 Analisis Hasil Pengujian

Setelah semua pengujian waktu eksekusi sistem dalam melakukan *task* dalam satu *loop* dengan menggunakan fungsi *Millis*. Dilakukan perhitungan ketepatan waktu eksekusi pada tiap metode. Didapatkan hasil uji masing-masing waktu eksekusi sistem yang dilakukan sebanyak 10 kali seperti yang terlihat pada Tabel 6.4 serta dilakukan perhitungan error dari ketepatan waktu eksekusi. Dengan Membandingkan waktu error eksekusi rata-rata hasil pada sistem pada satu siklus dibagi dengan pengukuran waktu yang nyata yang menggunakan *deadline* waktu selama yaitu 66 detik dengan menggunakan persamaan 2.1.

Hasil perhitungan error pada sistem *Millis* sebagai berikut.

$$\text{Nilai Persentase Error} = \frac{0,004}{66} \times 100\% = 0,006\%$$

Tabel 6.4 Data Waktu Eksekusi tiap loop sistem

Pengujian Loop ke	Waktu Eksekusi Program (s) dengan <i>deadline</i> (66s) Dalam Satu Loop dengan Menggunakan Millis
1	66
2	66
3	66
4	66
5	66,04
6	66
7	66
8	66
9	66
10	66
Rata-rata	66,004
Persentase Error Ketepatan Waktu Rata-rata	0,006%

Berdasarkan Tabel 6.4 Nilai rata-rata dari waktu eksekusi yang didapatkan pada sistem peneliti dengan menggunakan millis didapatkan 66,004 detik dan memiliki error waktu rata-rata sebesar 0,004 detik dengan persentasi error 0,006%.

Dapat disimpulkan penggunaan *delay* untuk melakukan *scheduling* task yang membutuhkan waktu yang eksekusi konstan, sangat tidak disarankan karena perhitungan delay hanya memberikan jeda pada tiap *task* yang dieksekusi tanpa memperhitungkan waktu yang telah dipakai sistem dalam menjalankan tiap *task*nya. Penjadwalan yang dilakukan pada Millis mampu memberikan eksekusi waktu yang lebih konstan dengan melakukan perhitungan waktu yang telah berjalan dikurangi waktu berjalan pada task sebelumnya dibandingkan dengan *interval* waktu untuk menjalankan *task*.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan berbagai analisis dari hasil yang diperoleh pada pengujian yang dilakukan pada penelitian ini, maka terdapat beberapa kesimpulan dari rumusan masalah yang ditentukan sebelumnya seperti yang akan dijelaskan sebagai berikut.

1. Untuk mengimplementasikan sistem yang real time atau sistem yang mampu melaksanakan respon tugas tepat batasan waktu yang ditentukan pada mikrokontroler, Fungsi Millis mampu memberikan solusi *multitasking* dan *scheduling* secara tepat sesuai dengan interval dan perhitungan waktu yang nyata pada saat sistem mulai dijalankan. Sistem mampu menyelesaikan seluruh *task* yaitu nyala lampu sekaligus pengaktifan pembacaan pelanggar dan pergerakan fokus kamera yang dijadwalkan dalam satu siklus selama 66 detik pada perhitungan waktu sistem sesuai dengan perhitungan nyata selama 66 detik yang telah diujikan.
2. Dalam pengujian pendeteksi sistem berhasil melakukan pendeteksian pelanggaran dengan menggunakan lima objek yang berbeda pada saat status blok lampu merah dalam keadaan aktif. Sistem mampu mengaktifkan buzzer sebagai alarm suara, melakukan *snapshot* serta berhasil menuliskan hasil data gambar ke dalam micro sd card pada saat pelanggaran terdeteksi. Berdasarkan pengujian waktu eksekusi *alarm* dilakukan dengan menggunakan tiga resolusi gambar yang berbeda pada lima objek yang berbeda, hasil rata-rata waktu eksekusi alarm yang didapatkan dengan menggunakan resolusi gambar 640x480 pixel adalah 19525 milli detik atau 19,52 detik, pada rata-rata waktu eksekusi dengan menggunakan resolusi 320x240 pixel adalah 5155,4 milli detik atau 5,15 detik dan pada rata-rata waktu eksekusi dengan menggunakan resolusi 160x120 pixel adalah 1580 milli detik atau 1,58 detik. Berdasarkan hasil pengujian yang dilakukan, penggunaan resolusi 320x240 lebih optimal digunakan pada sistem *prototype* peneliti yang mampu melakukan penyimpanan sekitar 5 detik dibandingkan dengan penggunaan resolusi 640x480pixel yang memakan waktu penyimpanan data selama 19 detik dan juga penggunaan resolusi 160x120pixel yang menghasilkan gambar dengan kualitas yang sangat rendah.
3. Pada pengujian ketepatan rata-rata waktu eksekusi program dengan menggunakan fungsi millis. Pada pengujian yang telah dilakukan yaitu dengan mengamati tiap siklus *loop* atau penjadwalan yang berjalan dengan *deadline* waktu 66 detik. Pada sistem peneliti mampu memberikan rata-rata waktu eksekusi selama 66,004 detik dan memiliki persentase waktu error sebesar 0,006%. Dapat disimpulkan penggunaan *delay* dalam melakukan *scheduling* dan *multitasking* untuk membangun sistem yang membutuhkan waktu yang penjadwalan eksekusi konstan atau *real time*, tidak disarankan

karena perhitungan delay hanya memberikan jeda pada tiap *task* yang dieksekusi tanpa memperhitungkan waktu yang telah dipakai sistem untuk menjalankan tiap *task*nya. Sehingga waktu dimulainya eksekusi *task* yang berikutnya mengalami kemunduran waktu selama waktu yang diperlukan oleh sistem untuk mengerjakan *task* sebelumnya ditambahkan dengan *delay* yang telah ditentukan pada sistem.

7.2 Saran

Agar sistem dapat dikembangkan lebih baik lagi, maka penulis akan memberikan beberapa saran sebagai berikut.

1. Melakukan pengujian dan membandingkan sistem *real time* pada mikro kontroler dengan menggunakan mikro komputer yang memiliki prosesor lebih dari satu seperti Raspberry Pi. Sehingga dalam melakukan *multitasking*, sistem mampu menjalankan *task*nya secara *concurrent* atau semua pekerjaan dapat bekerja secara bersamaan tanpa melakukan *multitasking* dengan menunggu pekerjaan lain selesai terlebih dahulu untuk menjalankan pekerjaan berikutnya atau disebut juga dengan *sequential*.
2. Untuk implementasi yang nyata, penulis tidak menyarankan untuk penggunaan modul kamera VC0706 dalam pengambilan data gambar yang hanya memiliki resolusi maksimal sebesar 640x480 pixel dan sekaligus menyarankan untuk menggunakan modul kamera ELP-USB8MP02G-75 yang mampu mengambil data gambar dengan resolusi maksimal sebesar 3264x2448 pixel.
3. Sistem dapat dikembangkan dalam mendefinisikan jenis pelanggaran yang terjadi maupun sistem mampu melakukan identifikasi pengguna kendaraan yang melakukan pelanggaran dengan memanfaatkan kamera.
4. Diharapkan pada penelitian selanjutnya, sistem mampu berjalan pada server secara online baik untuk melakukan pengontrolan sistem maupun digunakan untuk menyimpan data pada *database* yang mampu melakukan sinkronisasi antar blok lampu lintas dengan jarak yang cukup berjauhan.

DAFTAR PUSTAKA

- Alan, G. Smith., 2011. *Introduction to Arduino*
- Arduino Mega 2560 Specifications* [online] tersedia di: < <https://arduino.cc/en/Main/ArduinoBoardMega2560>> [Diakses 3 Maret 2018]
- Arduino Multitasking! How to Use Millis in Arduino Code* [online] tersedia di: < <https://www.allaboutcircuits.com/projects/arduino-millis-lesson/>> [Diakses 27 April 2018]
- Aruljothi, M., Ramya V., Palaniappan, B., 2012. *Embedded System For Automatic Traffic Violation Monitoring And Alerting*.
- Departemen Pekerjaan Umum, 1995. *Tata Cara Perencanaan Fasilitas Pejalan Kaki Di Kawasan Perkotaan*. Direktorat Bina Marga
- Departemen Pekerjaan Umum, 1997. *Manual Kapasitas Jalan Indonesia*. Direktorat Jendral Bina Marga.
- Earl, Bill., 2014. *Multi-tasking the Arduino* [online] tersedia di: < <https://www.cdn-learn.adafruit.com/downloads/pdf/multi-tasking-the-arduino-part1.pdf/>> [Diakses 27 April 2018]
- Gupta, Shambhu Kumar, Nand Kumar dan G. Tamizharasi., 2016. *Environment Monitoring and Device Control Using Arduino Based Embedded Control Sensor Network*. India: Dept. Of Electronics and Telecommunication Engineering.
- Hajeb, S., Javadi, M., Mashemi, S.M., 2013. *Traffic Violation Decetecion System based on RFID*. World Academy of Science, Engineering and Technology International Journal of Mechanical and Mechatronics Engineering.
- Hossain, M. R. T , Bhuiya,M. M. K , Ahamed, J. U , Bhuiyan, T. H , Bhowmik, S., 2010. *Monitoring And Reporting Of Traffic Rules Violation Using Microcontroller Through Wireless Communication System*. Bangladesh: Department of Electrical and Electronic Engineering Chittagong University of Engineering and Technology.
- Jatmiko, W., Mursanto, P., Jati, G., Purnomo, D. M., Alhamidi, M.R., Habibie, N dan Dwinto, K., 2015. *RTOS Teori dan Aplikasi*. Depok: Fakultas Ilmu Komputer Universitas Indonesia.
- John A. Stankovic, 1988. *Misconceptions about Real-Time Computing*. University of Massachusetts
- Jui, Sadia Afrose dan Borson Sarkan., 2016. *DESIGN AND DEVELOPMENT OF SMART SECURITY SYSTEM*. Bangladesh: East West University

- Kurniawan, F. R., Setiawan, I. dan Sumardi., 2012. *Multitasking Pada Mikrokontroler Atmega16 Menggunakan Real Time Operating System (RTOS) Jenis Cooperative*). Semarang: Universitas Diponegoro
- Mirzaei, Azar., 2013. *Checking Traffic Violation Using RFID*. International Journal of Computer Science & Network Solutions.
- Pantano, N., 2012. *Real Time Operating System on Arduino*.
- Sagar., 2002. *Embedded Operating Systems for Real-Time Applications*. Bombay: Indian Institute of Technology.
- SG90 9g Micro Servo Specifications [online] tersedia di: <akizukidenshi.com/download/ds/towerpro/SG90.pdf> [Diakses 3 Maret 2018]
- Suman, Singha dan Debasis Maji., 2016. *LASER SECURITY SYSTEM*. International Journal of Scientific & Engineering Research.
- TTL Serial Camera – Adafruit Industries [online] tersedia di: <cdn-learn.adafruit.com/downloads/pdf/ttl-serial-camera.pdf> [Diakses 3 Maret 2018]
- Undang – Undang No.22 Tahun 2009 Tentang Lalu Lintas Dan Angkutan Jalan (LLAJ) Pasal 131 Ayat 2
- Wongsokuncoro, Halim, Winarno, Rahmatillah, Akif., 2016. *Rancang Bangun Pendeteksian Pelanggaran Pada Traffic Light Berbasis Mikrokontroler*. Surabaya: Perpustakaan Universitas Airlangga